Real-Time Optimization via Modifier Adaptation of Closed-Loop Processes using Transient Measurements

Real-Time Optimization via Modifier Adaptation of Closed-Loop Processes using Transient Measurements

Jack Speakman*, Grégory François

School of Engineering, The University of Edinburgh, Edinburgh, EH9 3FB, UK

Abstract

Real-time optimization (RTO) has the ability to boost the performance of a process whilst satisfying the constraints by using process measurements, driving the operating conditions towards optimality. Modifier adaptation (MA) is a methodology of RTO which can find the optimal operating point of a process even in the presence of plant-model mismatch. This work presents an extension to MA through the combination of two established frameworks, allowing for the optimization of a controlled process using transient measurements whilst using a steady-state open-loop model. In addition, an approach for model-based gradient estimation, despite the mismatch between the degrees of freedom of the closed-loop plant and the available open loop model is suggested that does not necessitate amending the model. The proposed scheme is illustrated on a case study of a CSTR and a distillation column, detailing how the gradient can be estimated.

Keywords: Real-time optimization, Modifier adaptation, Transient measurements, Plant-model mismatch, Model-based gradient estimation

1. Introduction

Optimization of chemical processes is vital for improving their economic profile, whilst ensuring the meeting of safety and environmental objectives, in the face of increasing global competition and tightening regulations. Process optimization is typically carried out using a model to calculate the operating conditions which maximize the performance of the process. Fully accurate models are rarely obtained due to uncertainties, noise and simplifications, and also because it is experimentally costly; resulting in sub-optimal operation. Real-time optimization (RTO) solves this through the use of measurements from the process to adjust the operating point of the plant towards the true optimal solution.

There has been several different approaches suggested on how to use the measurements of the process to best adjust the operating conditions to increase the performance of the plant. The most widely implemented method is the two-step approach (Jang et al., 1987; Darby et al., 2011; Câmara et al., 2016). This approach uses the measurements to update a set of model parameters, and in turn uses this updated model to calculate a new set of operating conditions. This is iteratively repeated until convergence. Whilst this method is simple to understand, convergence to the optimum of the plant cannot be guaranteed in the presence of structural plant-model mismatch (Yip and Marlin, 2004).

For convergence to be at the plant optimum it is necessary that the Karush-Kuhn-Tucker (KKT) conditions of the model match those of the plant, which is satisfied if all the gradients of the objective and constraints of the model are equal to the plant at the converged operating point. This prompted the use of gradient information of the plant objective function in the two step approach (Roberts, 1979), resulting in a new approach labeled integrated system optimization and parameter estimation (ISOPE). This idea was further developed by Tatjewski (2002) to neglect the parameter estimation as this was not necessary for

*Corresponding Author

Email addresses: Jack.Speakman@ed.ac.uk (Jack Speakman), Gregory.Francois@ed.ac.uk (Grégory François)

convergence to the plant optimum. Gradient information of the constraints was then added to the approach (Gao and Engell, 2005), allowing for the KKT conditions to be satisfied at the current operating point. The resulting method, referred to as modifier adaptation (MA), does not modify the model parameters, rather the approach adds affine-in-input modifier terms directly to the cost and constraint functions, forcing the model to match the plant-based first order conditions of optimality (Marchetti et al., 2009).

The main advantage of MA lies in the mathematically proven capacity to converge to a KKT point of the process (Marchetti et al., 2009), even in the presence of structural plant-model mismatch. This advantage has sparked interest in the method and several variants have been developed which propose alternative formulations to improve upon the shortcomings of MA, or methods which extend its capabilities. One such limitation is the requirement of accurate plant gradients to estimate the modifiers. These gradients are generally not directly measured and must be estimated from measurements. Several approaches, some model-based, some others measurement-based (François et al., 2012) to gradient estimation have been outlined. Some proposed approaches to reduce the limitation of gradient estimation are through the use of previous measurements (Marchetti et al., 2010; Brdyś and Tatjewski, 1994; Gao and Engell, 2005), directional MA (Costello et al., 2016), nested MA (Navia et al., 2015), or through the use of quadratic approximations (Gao et al., 2016).

Typically, process optimization can only be implemented after extensive modeling of the process, which is often expensive and in certain circumstances not feasible to carry out, thus only an incomplete model which does not fully represent the real process may be available. As such, this incomplete model may be missing key information which not only results in plant-model mismatch, but could also result in the plant and model having a different set of degrees of freedom. This may be the case if the model has been developed with degrees of freedom which helped solve the model's complex system of equations, when in reality the plant has fewer decisions which can be made (Costello et al., 2013). Another example in which there may be a plant-model degree of freedom difference is for a process in which a control system has been put in place and an explicit form of the controller is unknown or is inconvenient to be included in the process optimization model. In both these cases, in order to carry out an optimization of the process using standard approaches, further modeling is required which as discussed previously may not be a readily available option due to the extensive time and cost considerations. An alternative to remodeling is to overcome this degree of freedom difference in the optimization scheme. Recently, an extension to MA based on this idea proposed three approaches which directly solve the plant optimization problem in the presence of a model with more degrees of freedom than the plant (François et al., 2016). These approaches are mathematically proven to reach a plant KKT point upon convergence, without the need to remodel the missing information.

The requirement of waiting until steady state and only using the data once the plant has settled, ignoring the transient measurements, may result in long convergence times. François and Bonvin (2014) suggested the use of these transient measurements, and updating the process inputs before the plant has settled, leading to convergence in a single iteration to steady state. However, this approach is limited to cases whereby the degrees of freedom of the plant and the model are the same, i.e. if both are open- or closed-loop. In this article it is shown that transient RTO via modifier adaptation can be applied when the plant and the model do not share the same inputs, or when the plant operates in closed-loop while only an open-loop model is available, without remodeling, even if model-based gradient techniques are used to estimate the gradients of the plant.

This article is structured as follows, Section 2 will state the mathematical optimization problem, with a brief overview of MA, and Section 3 presents the controlled process and transient measurement extensions in further detail. This is followed a detailed discussion of the newly proposed scheme of using transient measurements in the optimization of a controlled process in Section 4, along with a proposition of the KKT converging nature of the combined framework and associated proof for each method. In Section 5, an analysis of the gradients involved in the proposed method and why KKT convergence is observed with known plant gradients is presented. This is backed up with a further investigation into the use of model-based gradient estimation methods for a controlled process is provided along with a proposition on how model-based gradient estimation techniques can be implements for a controlled plant with a change in the degrees of freedom. Finally, this new framework will be demonstrated firstly in a case study of a CSTR in Section 6, then on a distillation column in Section 7.

2. Problem Definition

2.1. Process Optimization

The aim of process optimization is to find a set of operating conditions which minimize (or maximize) an objective function, indicating the cost (or profit) associated with the process. This set of conditions is also required to satisfy a set of constraints on the process, therefore the plant-based optimization can be written as the solution to a non-linear problem (NLP):

$$\begin{aligned} \boldsymbol{u}_p^* \coloneqq \arg\min_{\boldsymbol{u}} \quad & \phi_p(\boldsymbol{u}) \coloneqq \Phi(\boldsymbol{u}, \boldsymbol{y}_p(\boldsymbol{u})), \\ \text{s.t.} \quad & \boldsymbol{g}_p(\boldsymbol{u}) \coloneqq \boldsymbol{G}(\boldsymbol{u}, \boldsymbol{y}_p(\boldsymbol{u})) \leq \boldsymbol{0}, \end{aligned} \tag{1}$$

where $\boldsymbol{u} \in \mathbb{R}^{n_u}$ are the open-loop process input variables, $\boldsymbol{y}_p \in \mathbb{R}^{n_y}$ are the process output variables, $\phi_p \in \mathbb{R}$ and $\boldsymbol{g}_p \in \mathbb{R}^{n_j}$ are the plant based cost and constraint functions respectively. The subscript $(\cdot)_p$ indicates quantities and/or properties related to the plant. The mathematical form of the cost and constraint functions is assumed to be known for a given set of inputs and measured outputs, however the output function, $\boldsymbol{y}_p(\boldsymbol{u})$, for the plant is assumed to be unknown, therefore this NLP cannot be solved directly. In order to calculate the optimal conditions of the plant, a process model at steady state is generally used:

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) = \boldsymbol{0}, \tag{2}$$

$$\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}), \tag{3}$$

where $\boldsymbol{x}$ are the state variables at steady state and $\boldsymbol{\theta}$ are the model parameters. Equation (2) can be rearranged for $\boldsymbol{x}$ ($\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta})$) and substituted into Equation (3) to get:

$$\boldsymbol{y}(\boldsymbol{u}, \boldsymbol{\theta}) \coloneqq \boldsymbol{H}(\boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}), \boldsymbol{u}, \boldsymbol{\theta}). \tag{4}$$

This model can be used to formulate a model-based optimization problem to approximate the optimal solution of the plant:

$$\begin{aligned} \boldsymbol{u}^* \coloneqq \arg\min_{\boldsymbol{u}} \quad & \varphi(\boldsymbol{u}, \boldsymbol{\theta}) \coloneqq \Phi(\boldsymbol{u}, \boldsymbol{y}(\boldsymbol{u}, \boldsymbol{\theta})), \\ \text{s.t.} \quad & \mathcal{G}(\boldsymbol{u}, \boldsymbol{\theta}) \coloneqq \boldsymbol{G}(\boldsymbol{u}, \boldsymbol{y}(\boldsymbol{u}, \boldsymbol{\theta})) \leq \boldsymbol{0}, \end{aligned} \tag{5}$$

with the different scripture of the cost and constraint functions indicating different formulations with respect to the different function arguments. In the presence of plant-model mismatch, i.e. $\boldsymbol{y}(\boldsymbol{u}) \neq \boldsymbol{y}_p(\boldsymbol{u})$, the optimal solution to the model-based optimization problem is not guaranteed to match the plant optimum, leading to suboptimal performance, or even infeasible operation. This observation has led to the use of plant measurements, along with the erroneous model, to modify the operating conditions of the plant towards optimality through real-time optimization (RTO). For example, with the two-step approach, the parameters of the model are iteratively updated and the updated model is used to calculate the new operating point. However, this method is only guaranteed to converge to a KKT point of the plant if there exists a value of $\boldsymbol{\theta}$ such that the optimum defined by (5) matches that defined by (1), i.e. when the real plant belongs to the set of models obtained when $\boldsymbol{\theta}$ varies.

2.2. Standard Modifier Adaptation

Another approach to RTO is modifier adaptation (MA), which adds affine-in-input correction terms to the cost and constraints. These modifiers are chosen to force the modified model to match the first-order conditions of optimality of the plant.

$$\begin{aligned} \boldsymbol{u}_{k+1}^* \coloneqq \arg\min_{\boldsymbol{u}} \quad & \varphi_k(\boldsymbol{u}, \boldsymbol{\theta}), \\ \text{s.t.} \quad & \mathcal{G}_k(\boldsymbol{u}, \boldsymbol{\theta}) \leq \boldsymbol{0}, \end{aligned} \tag{6}$$

where φ_k and $\mathcal{G}_k$ are the modified cost and constraints functions, respectively. These modified functions are defined as:

$$\varphi_k(\boldsymbol{u}, \boldsymbol{\theta}) \coloneqq \varphi(\boldsymbol{u}, \boldsymbol{\theta}) + \varepsilon_k^{\phi} + (\boldsymbol{\lambda}_k^{\phi})^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k), \tag{7}$$

$$\mathcal{G}_k(\boldsymbol{u}, \boldsymbol{\theta}) \coloneqq \mathcal{G}(\boldsymbol{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k^{g} + (\boldsymbol{\lambda}_k^{g})^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k), \tag{8}$$

with the zeroth order modifiers (ε), and the first order modifiers (λ) defined as:

$$\varepsilon_k^{\phi} = \phi_p(\boldsymbol{u}_k) - \varphi(\boldsymbol{u}_k, \boldsymbol{\theta}), \tag{9}$$

$$\boldsymbol{\varepsilon}_k^{g} = \boldsymbol{g}_p(\boldsymbol{u}_k) - \mathcal{G}(\boldsymbol{u}_k, \boldsymbol{\theta}), \tag{10}$$

$$\boldsymbol{\lambda}_k^{\phi} = \nabla_{\boldsymbol{u}}\phi_p(\boldsymbol{u}_k) - \nabla_{\boldsymbol{u}}\varphi(\boldsymbol{u}_k, \boldsymbol{\theta}), \tag{11}$$

$$\boldsymbol{\lambda}_k^{g} = \nabla_{\boldsymbol{u}}\boldsymbol{g}_p(\boldsymbol{u}_k) - \nabla_{\boldsymbol{u}}\mathcal{G}(\boldsymbol{u}_k, \boldsymbol{\theta}), \tag{12}$$

where $\nabla_{\boldsymbol{u}}(\cdot)$ denotes the gradient operator w.r.t. $\boldsymbol{u}$. The zeroth order cost modifier is usually neglected as this does not change the location of the solution, but only shifts the cost function up or down . As can be seen, the measurements of the plant are used to estimate the steady-state plant values and gradients of the cost and constraints at the current operating point, given as $\phi_p(\boldsymbol{u}_k)$, $\boldsymbol{g}_p(\boldsymbol{u}_k)$, $\nabla_{\boldsymbol{u}}\phi_p(\boldsymbol{u}_k)$ and $\nabla_{\boldsymbol{u}}\boldsymbol{g}_p(\boldsymbol{u}_k)$. Additionally, the modifiers can be filtered via an exponential filter, for example for the cost zeroth order modifier, $\varepsilon_k^{\phi} = (\boldsymbol{I} - \boldsymbol{K})\varepsilon_{k-1}^{\phi} + \boldsymbol{K}\left(\phi_p(\boldsymbol{u}_k) - \varphi(\boldsymbol{u}_k, \boldsymbol{\theta})\right)$.

Algorithm 1 MA Algorithm with filtering of the modifiers

Begin
Initialize all modifiers to $\mathbf{0}$
Modify the model-based optimization problem (6)
for $k = 0 \rightarrow \infty$
1: Solve modified model-based optimization problem and deduce the inputs to apply to the plant
2: Implement plant inputs
3: Take measurements of the plant
4: Estimate plant gradients
5: Calculate the modifiers
6: Apply exponential filter
7: Reformulate modified model-based problem
end

The general algorithm for MA is stated in Algorithm 1, with all of the extensions discussed in this article following the same approach but with different optimization problems. As the model parameters are assumed to be fixed, they can be neglected from the problem formulation, resulting in:

$$\phi(\boldsymbol{u}) \coloneqq \varphi(\boldsymbol{u}, \boldsymbol{\theta}_n), \tag{13}$$

$$\boldsymbol{g}(\boldsymbol{u}) \coloneqq \mathcal{G}(\boldsymbol{u}, \boldsymbol{\theta}_n), \tag{14}$$

where $\boldsymbol{\theta}_n$ is the nominal parameters to the model. The main advantage of this method is that, upon convergence, if the gradients of the plant are accurately estimated, this method is guaranteed that the only point which can be converged to is a KKT point of the plant. This is due to the modifiers matching the local first order properties of the model with the plant, therefore upon convergence, as the operating point is a KKT point of the model, it must also be a KKT point of the plant (Marchetti et al., 2009).

2.3. Gradient Estimation

The effectiveness of MA is highly dependent on the accuracy of the gradient estimates of the plant, with convergence to the plant optimum only occurring with perfect gradient estimation. Estimation of the plant

gradients is often a trade-off between time (or experimental cost) and accuracy. A summary of different approaches is given hereafter.

Finite Difference. The most common method for MA is to use a finite differences approach, which uses n_u perpendicular perturbations from the current operating point, then calculates the gradient by dividing the difference between the perturbed and nominal values by the size of the perturbation,

$$\nabla_{\boldsymbol{u}}\phi_p \approx \frac{\phi_p(\boldsymbol{u}+\delta\boldsymbol{u})-\phi_p(\boldsymbol{u})}{\delta\boldsymbol{u}}, \tag{15}$$

where $\delta\boldsymbol{u}$ is the perturbation. With a good state estimator to remove noise and accurately estimate the cost and constraints for a given set of inputs, this method can lead to good estimates for the gradient. However, the number of steady state to steady state iterations for the estimation of the plant gradient increases linearly with n_u and therefore is potentially very expensive and time consuming.

Dual Approach. This approach uses current and past measurements to estimate the gradients (Marchetti et al., 2010; Gao and Engell, 2005; Brdyś and Tatjewski, 1994), taking advantage of the automatic perturbation caused by the RTO algorithm. The method used to calculate the gradient is to use the previous n_u+1 measurements to fit a plane for the objective and constraint functions and use this fit for the gradient estimate. A trade-off between iterations which give good gradient estimates and improved operating conditions is required, resulting in a dual objective. This can be solved though a constraint which limits the step size and direction, favoring the gradient estimation, whilst maintaining an objective function which minimizes the cost.

Neighboring Extremals (NE). This approach is a model-based method which uses the parametric sensitivity of the model to estimate the gradient of the plant (Gros et al., 2009). This method exploits the difference in the outputs of the model and the plant, to estimate the difference in the gradient of the plant and model.

$$\begin{aligned}\nabla_{\boldsymbol{u}}\phi_p \approx &\nabla_{\boldsymbol{u}}\varphi(\boldsymbol{u}_0,\boldsymbol{\theta}) + \nabla^2_{\boldsymbol{u\theta}}\varphi(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+\delta\boldsymbol{y}_p + \\ &(\nabla^2_{\boldsymbol{uu}}\varphi - \nabla^2_{\boldsymbol{u\theta}}\varphi(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+\nabla_{\boldsymbol{u}}\boldsymbol{H})\delta\boldsymbol{u}_p,\end{aligned} \tag{16}$$

where $\delta\boldsymbol{y}_p$ is the difference between the measured outputs of the plant and the nominal model, $\delta\boldsymbol{u}_p$ is the difference between plant and nominal operating points, and $\boldsymbol{H}(\boldsymbol{u},\boldsymbol{\theta})$ is the measurable output function of estimated states of the model, defined by Equation (4). This approach requires more measured outputs than model parameters, and as the parametric sensitivity is used, the accuracy of this method depends on the accuracy of the model, with good estimates in the case of parametric and minimal structural uncertainties.

Modifier Adaptation with Quadratic Approximation (MAWQA). This approach uses a distribution of local and distant operating points to regress a quadratic approximation of the cost and constraint functions (Gao et al., 2016). These quadratic approximations can be used to estimate the gradient of the plant at the current operating point, reducing the number of steady-state measurements required at each iteration, whilst maintaining an accurate gradient estimate.

Other Approaches. There are several other approaches which have been used in literature (François et al., 2012). These include the use of multiple units (MU) to calculate the gradient via the finite difference method but in a single steady-state convergence by having several identical units in parallel and offsetting each from the current nominal operating point (Srinivasan, 2007), but having several identical plants in parallel can be rare in practice.

2.4. Controlled Process Problem Formulation

It has been assumed up to now that the inputs to the plant and model have the same degrees of freedom, $\boldsymbol{u}$. This is not the case for a process for which a model has been developed in open-loop formation, i.e. with $\boldsymbol{u}$ as the degrees of freedom, while the plant is equipped with an external control scheme, whose set points $\boldsymbol{r}$ are indeed the plant degrees of freedom. This is illustrated in Figure 1. The NLP for a controlled process can be formulated as follows,

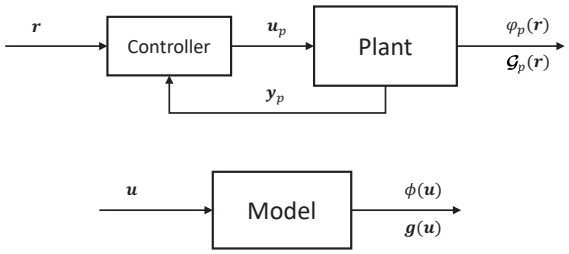


Figure 1: Illustration of flowchart of degree of freedom difference between a controlled plant and open-loop model

$$\begin{aligned} \boldsymbol{r}^* \coloneqq \arg\min_{\boldsymbol{r}} \quad & \varphi_p(\boldsymbol{r}) \coloneqq \Phi\left(\boldsymbol{u}_p(\boldsymbol{r}), \boldsymbol{y}_p\left(\boldsymbol{u}_p(\boldsymbol{r})\right)\right), \\ \text{s.t.} \quad & \boldsymbol{\mathcal{G}}_p(\boldsymbol{r}) \coloneqq \boldsymbol{G}\left(\boldsymbol{u}_p(\boldsymbol{r}), \boldsymbol{y}_p\left(\boldsymbol{u}_p(\boldsymbol{r})\right)\right) \leq \boldsymbol{0}. \end{aligned} \tag{17}$$

Standard MA requires for the degrees of freedom, i.e. the decision variables, to be the same between the model and the plant. Therefore, in the case of having a controlled plant, where the degrees of freedom of this system are the set points to the controller ($\boldsymbol{r}$); if an explicit form of the controller is not known by the optimization algorithm, standard MA cannot be directly applied.

3. Recent Extensions to MA

This article will propose an approach which allows for the optimization of a controlled process using a steady-state open-loop model and transient measurements. This allows for a rapid approach to the optimal conditions despite having a difference in degrees of freedom between the plant and the model. In order to do this, two frameworks are introduced as a basis of this approach. The first proposing an approach to solve the degree of freedom difference (but that only use steady-state measurements), whilst the second proposing a method of utilizing transient measurements (but only when the plant and the model share the same inputs).

3.1. Closed-Loop MA with Open-Loop Model

Optimization of processes provides an opportunity for a plant to improve the operating profit, whilst ensuring safe working conditions and environmental regulations are met. Despite the great claims of process optimization, advanced optimization schemes such as RTO are rarely implemented in practice (Quelhas et al., 2013). There may be several reasons for this, one may be that there is a large time delay between the cutting edge in academia and what is used in industry. This time delay allows for the newly developed approaches to be refined until they are tried and tested and proven to provide clear improvements with limited risk. Another potential reason for the lack of willingness for the cutting edge ideas to be taken up is that often the cases dealt with are idealized situations with little regard for what the true situations are in practice (Darby et al., 2011). One condition, which is commonly assumed is that the model at hand is a good, if not ideal, representation of the plant to optimize. Additionally, if this condition is not met, it is assumed that it is possible to rectify the situation by remodeling the model used in the optimization scheme by incorporating the details of the plant which may be missing, once they are known.

This article focuses the cases whereby, on top of parametric and structural plant-model mismatch, the model and the plant do not share the same inputs. There are several scenarios in which the degrees of freedom may be inconsistent between the plant and model, for example if the model has been developed for process design where the temperatures, pressures, flow rates, and other conditions are optimized to find the best piece of equipment which satisfies these conditions, but once implemented, the degrees of freedom of this unit are not as free as the original model. Alternatively, the process model may be developed in open-loop before the design for a control system has been implemented.

In these cases, changing the model used in the optimization problem is required to be able to use standard RTO techniques. If the model and the qualified personal are available to implement this remodeling, then this is a viable approach, if not the best (since the quality of the model is improved while the operation of the plant is optimized). However, in practice there are many scenarios in which is may not be a viable or convenient option, e.g. only a complied version of the coding being available.

If it is the case that the model cannot be easily reformulated into the same degrees of freedom of the plant, the only option for improving the productivity of the process would be limited to using approaches which do not utilize a model. The necessity of requiring a model with the same degrees of freedom as the plant was put into question with a new MA framework (François et al., 2016) which directly solves this degree of freedom difference through three different proposed approaches. All three approaches use the model-based outputs $\boldsymbol{y}_r(\boldsymbol{u})$ which correspond to the set points of the plant as an estimate of the optimal conditions of the plant, i.e. $\boldsymbol{r}_k = \boldsymbol{y}_r(\boldsymbol{u}_k)$. These approaches differ in the form of the modifiers, either solved in $\boldsymbol{r}$ or $\boldsymbol{u}$, and in the basis of the optimization algorithm. The following briefly describes how the model problem can be redefined to solve the controlled plant problem, given by Equation (17).

Method UR. The first method solves the optimization in $\boldsymbol{u}$ and the modifiers in $\boldsymbol{r}$.

$$\begin{aligned} \boldsymbol{u}_{k+1} \coloneqq \arg\min_{\boldsymbol{u}} \quad & \phi_k^r(\boldsymbol{u}), \\ \text{s.t.} \quad & \boldsymbol{g}_k^r(\boldsymbol{u}) \leq \boldsymbol{0}, \end{aligned} \tag{18}$$

where the modified functions are defined as,

$$\begin{aligned} \phi_k^r(\boldsymbol{u}) &\coloneqq \phi(\boldsymbol{u}) + (\boldsymbol{\lambda}_k^{\phi,r})^{\mathsf{T}}(\boldsymbol{y}_r(\boldsymbol{u}) - \boldsymbol{r}_k), \\ \boldsymbol{g}_k^r(\boldsymbol{u}) &\coloneqq \boldsymbol{g}(\boldsymbol{u}) + \boldsymbol{\varepsilon}_{k-1}^g + (\boldsymbol{\lambda}_k^{g,r})^{\mathsf{T}}(\boldsymbol{y}_r(\boldsymbol{u}) - \boldsymbol{r}_k). \end{aligned} \tag{19}$$

The linear modifiers, $\boldsymbol{\lambda}^r$, are calculated with respect to the plant set points, therefore the gradient of the model w.r.t. $\boldsymbol{r}$ is required. This can be achieved by multiplying the gradient by the inverse of the mapping between the respective outputs and the model inputs,

$$\boldsymbol{\lambda}_k^{\phi,r} \coloneqq \nabla_{\boldsymbol{r}}\varphi_p - \nabla_{\boldsymbol{u}}\phi \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}\right)^+, \tag{20}$$

$$\boldsymbol{\lambda}_k^{g,r} \coloneqq \nabla_{\boldsymbol{r}}\boldsymbol{\mathcal{G}}_p - \nabla_{\boldsymbol{u}}\boldsymbol{g} \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}\right)^+, \tag{21}$$

where $(\cdot)^+$ is the Moore-Penrose pseudoinverse.

Method UU. This method differs from method UR in that the modifiers are solved w.r.t. the process inputs $\boldsymbol{u}$, rather than the plant set points $\boldsymbol{r}$. This formulation can be defined by remapping the previously defined modifiers for method UR, given by Equations (20) and (21).

$$\boldsymbol{\lambda}_k^{\phi,u} \coloneqq \nabla_{\boldsymbol{r}}\varphi_p \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}\right) - \nabla_{\boldsymbol{u}}\phi \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}\right)^+ \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}\right), \tag{22}$$

$$\boldsymbol{\lambda}_k^{g,u} \coloneqq \nabla_{\boldsymbol{r}}\boldsymbol{\mathcal{G}}_p \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}\right) - \nabla_{\boldsymbol{u}}\boldsymbol{g} \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}\right)^+ \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}\right). \tag{23}$$

This allows for the formulation of a new optimization algorithm which directly solves both the optimization and affine modification in $\boldsymbol{u}$,

$$\begin{aligned} \boldsymbol{u}_{k+1} \coloneqq \arg\min_{\boldsymbol{u}} \quad & \phi_k^u(\boldsymbol{u}), \\ \text{s.t.} \quad & \boldsymbol{g}_k^u(\boldsymbol{u}) \leq \boldsymbol{0}, \end{aligned} \tag{24}$$

where the modified functions are defined as:

$$\phi_k^u(\boldsymbol{u}) \coloneqq \phi(\boldsymbol{u}) + (\boldsymbol{\lambda}_k^{\phi,u})^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k), \tag{25}$$

$$\boldsymbol{g}_k^u(\boldsymbol{u}) \coloneqq \boldsymbol{g}(\boldsymbol{u}) + \boldsymbol{\varepsilon}_k^g + (\boldsymbol{\lambda}_k^{g,u})^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k). \tag{26}$$

Method RR. Finally, the last method uses the model to create a convex[1] mapping between the outputs of the model and the cost and constraints. This convex remapping allows for a function to directly solve the model-based cost and constraints directly w.r.t. the set points of the plant, thus rectifying degree of freedom difference between the plant and the model and standard MA can be applied.

$$\begin{aligned} \boldsymbol{r}_{k+1} \coloneqq \arg\min_{\boldsymbol{r}} \quad & \varphi_k^c(\boldsymbol{r}), \\ \text{s.t.} \quad & \boldsymbol{\mathcal{G}}_k^c(\boldsymbol{r}) \leq \boldsymbol{0}, \end{aligned} \tag{27}$$

where the modified functions are defined as,

$$\begin{aligned} \varphi_k^c(\boldsymbol{r}) &\coloneqq \varphi^c(\boldsymbol{r}) + (\boldsymbol{\lambda}_k^{\phi,r})^\mathsf{T}(\boldsymbol{r} - \boldsymbol{r}_k), \\ \boldsymbol{\mathcal{G}}_k^c(\boldsymbol{r}) &\coloneqq \boldsymbol{\mathcal{G}}^c(\boldsymbol{r}) + \boldsymbol{\varepsilon}_k^g + (\boldsymbol{\lambda}_k^{g,r})^\mathsf{T}(\boldsymbol{r} - \boldsymbol{r}_k), \end{aligned} \tag{28}$$

where φ^c and $\boldsymbol{\mathcal{G}}^c$ are the convex mappings between the set point variables and the cost and constraints of the model. With the linear modifiers defined as,

$$\boldsymbol{\lambda}_k^{\phi,r} \coloneqq \nabla_{\boldsymbol{r}}\varphi_p - \nabla_{\boldsymbol{r}}\varphi^c, \tag{29}$$

$$\boldsymbol{\lambda}_k^{g,r} \coloneqq \nabla_{\boldsymbol{r}}\boldsymbol{\mathcal{G}}_p - \nabla_{\boldsymbol{r}}\boldsymbol{\mathcal{G}}^c. \tag{30}$$

With the optimization problems defined for each method, all three methods follow a similar algorithm to standard MA but using the newly defined problems respectively. As with standard MA, exponential filters can be applied to the modifiers to increase the conservativeness of each iteration for all three approaches. These approaches all solve the issue of a degree of freedom difference by combining the unknown controller with the open-loop plant and treating this as extra plant-model mismatch, resulting in a model which may poorly predict the controlled plant. Importantly, this new formulation does not require remodeling the original model, thus if for what ever reason reformulating the NLP of the model into the same formulation as the plant, i.e. into closed loop, is not desired, it can be avoided. This additional mismatch is not an issue with the MA algorithm (assuming accurate plant gradients), and still guarantees KKT matching upon convergence, with the associated proofs provided in the original paper (François et al., 2016). Convergence to the KKT point is dependent on the accuracy of the plant gradients, which for this case are assumed to be perfect as the approach uses static measurements with a finite difference approach.

3.2. Use of Transient Measurements

Another limitation of the standard MA formulation is that steady-state is required to be achieved between iterations. In the case of processes which have slow dynamics, this can result in long convergence times. Further, if the process has rapidly changing schedules or unknown disturbances which change the optimal operating conditions, slow RTO convergence times can result in poor performance. This prompted the use of measurements during the transient, before the process has reached steady-state. These measurements can be used to update the plant more frequently, resulting in faster convergence times and convergence in a single smooth step, rather than several steady-state iterations. One such method (François and Bonvin, 2014) of using these transient measurements is to directly use the measurements as estimates of the steady-state values, $\hat{\boldsymbol{y}}_p = \boldsymbol{y}_p(t)$, where $\hat{\cdot}$ refers to the steady-state estimate. The optimization problem can be defined as:

$$\begin{aligned} \boldsymbol{u}_{k+1} \coloneqq \arg\min_{\boldsymbol{u}} \quad & \phi_k(\boldsymbol{u}), \\ \text{s.t.} \quad & \boldsymbol{g}_k(\boldsymbol{u}) \leq \boldsymbol{0}, \end{aligned} \tag{31}$$

[1]The fact that the mapping is convex is an additional feature which enforces the model adequacy condition, as the resulting approximate model is forced to be convex everywhere, it will also be at the unknown plant optimum (François and Bonvin, 2013)

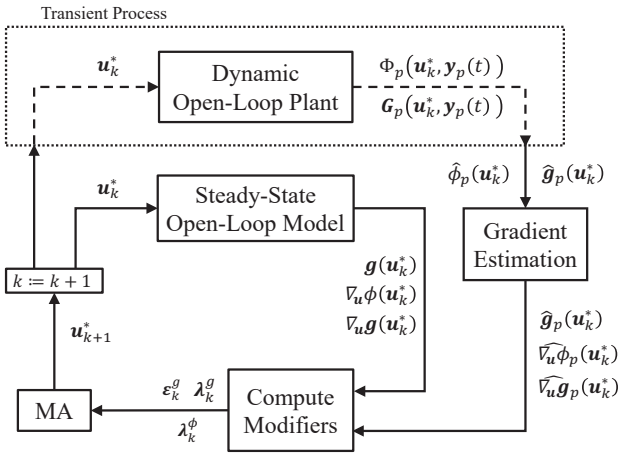


Figure 2: Illustration of the use of transient measurements framework

where the modified functions are defined as,

$$\phi_k(\boldsymbol{u}) \coloneqq \phi(\boldsymbol{u}) + (\widehat{\boldsymbol{\lambda}}_k^{\phi})^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k), \tag{32}$$

$$\boldsymbol{g}_k(\boldsymbol{u}) \coloneqq \boldsymbol{g}(\boldsymbol{u}) + \hat{\boldsymbol{\varepsilon}}_k^{g} + (\widehat{\boldsymbol{\lambda}}_k^{g})^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k). \tag{33}$$

The modifiers are defined w.r.t. the estimated gradients $\widehat{\nabla}_{\boldsymbol{u}}\phi_p$ and $\widehat{\nabla}_{\boldsymbol{u}}\boldsymbol{g}_p$:

$$\hat{\boldsymbol{\varepsilon}}_k^{g} \coloneqq \hat{\boldsymbol{g}}_p(\boldsymbol{u}_k) - \boldsymbol{g}(\boldsymbol{u}_{\boldsymbol{k}}), \tag{34}$$

$$\widehat{\boldsymbol{\lambda}}_k^{\phi} \coloneqq \widehat{\nabla}_{\boldsymbol{u}}\phi_p - \nabla_{\boldsymbol{u}}\phi, \tag{35}$$

$$\widehat{\boldsymbol{\lambda}}_k^{g} \coloneqq \widehat{\nabla}_{\boldsymbol{u}}\boldsymbol{g}_p - \nabla_{\boldsymbol{u}}\boldsymbol{g}, \tag{36}$$

where $\hat{\boldsymbol{g}}_p(\boldsymbol{u}_k)$ is the estimate of the steady-state constraint, which can be approximated by the transient measurement, $\hat{\boldsymbol{g}}_p(\boldsymbol{u}_k) = \boldsymbol{G}_p(\boldsymbol{u}_k, \hat{\boldsymbol{y}}_p)$, as illustrated in Figure 2. If the estimates for the gradients are correct upon convergence, then this method guarantees the same convergence properties of standard MA, as once the algorithm has converged the transient modified problem will be equivalent to standard MA modified problem. As these measurements are taken during the transient, the standard gradient estimation method of finite differences cannot be applied. François and Bonvin (2014) has suggested the use of model-based gradient estimation methods such as NE, or through the use of MU, which is investigated further in this article. Ferreira et al. (2017) suggested an extension to the use of steady state models, taking advantage of a dynamic model to produce better estimates of the steady state from the transient measurements. This is of particular importance for cases where there may be a response in the opposite direction to the true plant steady state point.

As an alternative to a model-based gradient estimation, Rodríguez-Blanco et al. (2017) suggested the use of a dual approach using previous transient measurements to estimate the gradient. This approach offers an alternative method which utilizes the same approach as the UR method discussed above and uses the measurements to directly solve the gradient of the plant in the degrees of freedom of the plant. Directly using the gradient estimation approach suggested by Marchetti et al. (2010) of using the previous $n_u + 1$ measurements results in a less stable approach, which prompted the use of a gradient estimation method based around an adaptive filtering algorithm (Vahidi et al., 2005), labeled the recursive extended least squares algorithm (RELS), applied to MA. This approach treats the gradients as a parameter of a quadratic model and continually fits these using the previous measurements, whilst forgetting the previous

measurements with an exponential filter, thus is less sensitive to the use of transient measurements. The gradient estimation approach is summarized as follows, initially formulating a Taylor series approximation of the plant.

$$\widehat{\Delta\varphi}_k = \boldsymbol{\beta}_{k-1}^{\mathsf{T}} \hat{\boldsymbol{\theta}}_{k-1}, \tag{37}$$

$$\boldsymbol{\beta}_{k-1}^{\mathsf{T}} = \left[\Delta \boldsymbol{r}_{k-1}^{\mathsf{T}}, \frac{1}{2} \Delta \boldsymbol{r}_{k-1} \Delta \boldsymbol{r}_{k-1}^{\mathsf{T}} \right], \tag{38}$$

$$\hat{\boldsymbol{\theta}}_{k-1}^{\mathsf{T}} = \left[\nabla_r \varphi, \nabla_r^2 \varphi \right]_{\boldsymbol{r}_{k-1}}, \tag{39}$$

where $\Delta \boldsymbol{r}_{k-1} = \boldsymbol{r}_{k-1} - \boldsymbol{r}_{k-2}$. From here, the gradient terms, given by $\hat{\boldsymbol{\theta}}_{k-1}^{\mathsf{T}}$, can be treated as parameters and approximated using the following,

$$\boldsymbol{\zeta}_{k-1} = \frac{1}{\gamma} \left(\boldsymbol{\Sigma}_{k-1} \right)^{-1} \boldsymbol{\beta}_{k-1} \left(1 + \frac{1}{\gamma} \boldsymbol{\beta}_{k-1}^{\mathsf{T}} \left(\boldsymbol{\Sigma}_{k-1} \right) \boldsymbol{\beta}_{k-1} \right)^{-1}, \tag{40}$$

$$\left(\boldsymbol{\Sigma}_k \right)^{-1} = \frac{1}{\gamma} \left(\boldsymbol{\Sigma}_{k-1} \right)^{-1} \left(1 - \boldsymbol{\zeta}_{k-1} \boldsymbol{\beta}_{k-1}^{\mathsf{T}} \right), \tag{41}$$

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{\zeta}_{k-1} e_k, \tag{42}$$

where $e_k = \Delta\varphi_p - \widehat{\Delta\varphi}_k$ is the difference between the actual observed change in the objective function and the predicted change, $\boldsymbol{\Sigma}_k$ is the covariance matrix of the estimated error, and γ is the forgetting factor. The same procedure can be applied to the constraints to estimate their gradients.

As this approach uses the previous measurements as a basis of the gradient calculation, a persistent excitation of the plant inputs is required such as the inverse of the condition number, $\kappa^{-1}(\boldsymbol{S}_k) \geq a$ (Brdyś and Tatjewski, 1994), where $\boldsymbol{S}_k$ represents the normalized difference between the current input and the n_u most recent inputs. This approach is compared to the method proposed in this article in Section 6.

Several other approaches which take advantage of the use of transient measurements, most of which utilize previous measurements as the basis for the estimation of the steady state gradient at the current operating point. These include, but are not limited to, Dynamic MAWQA (Gao and Engell, 2016; Cadavid et al., 2017) which utilizes the MAWQA framework with steady state estimation techniques during the transient to improve convergence rates.

Static approaches to RTO have the advantage of using the abundance of measurements to filter out noise from the estimates, thus providing some resistance to random noise in the system. When using transient measurements, this is typically more difficult to achieve, resulting in a strong influence from noise in the system. This is of particular importance for measurement based gradient estimators as if the previous data is not reliable, this can cause major error in the estimate of the gradient, thus the importance of sufficient excitation to reduce this impact. In terms of model-based gradient estimation methods such as NE, the difference between the nominal and converged state is used in estimating the true plant gradient, as described by Equation (16). Thus, if the nominal and converged points are sufficiently far apart, the impact of this noise will be greatly reduced.

4. A new combined MA framework for the Optimization of Controlled Processes using an Open-loop Model and Transient Measurements

The two frameworks discussed above are combined into a single framework, allowing for the use of transient measurements in the optimization of a controlled process using an open-loop steady-state model. The proposed method is based on using the transient measurements to estimate the steady-state value and gradients of the plant at the current operating point. These estimates can be used in the methods UR, UU and RR to solve for the next operating point. This section will outline the algorithms undertaken by each method followed by a proposition of the KKT point convergence and associated proofs for each of the three

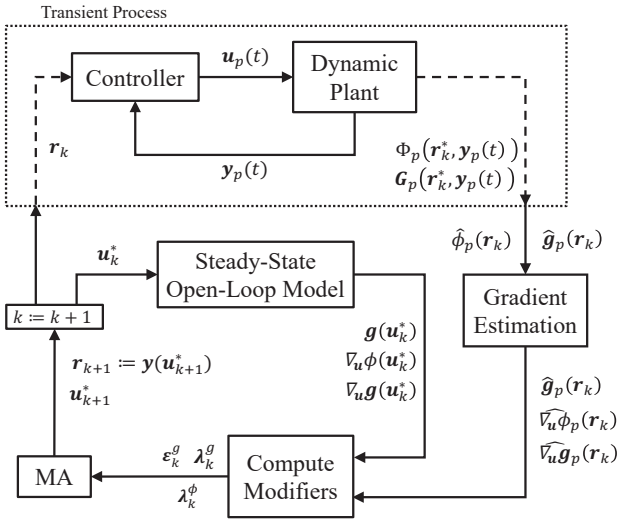


Figure 3: Illustration of combined flowchart for methods UR and UU. For Method RR, the open-loop model operates directly in $\boldsymbol{r}_k$ which are calculated directly from the MA scheme, and the outputs are also w.r.t. the controller set points

methods. The associated proofs of KKT matching upon convergence for these methods assume that the controlled plant is completely unknown, including the control scheme, however accurate gradient estimates of the controlled plant are available. In Section 5.2 this assumption of accurate plant gradient estimates with an unknown plant is reviewed and the assumption of an unknown control scheme is relaxed in order to obtain more accurate estimates using limited knowledge of the control scheme when using model based gradient estimation approaches.

4.1. Method UR - Transient

The following outlines the method UR using transient measurements. Firstly, the optimization problem to be solved can be defined as follows:

$$\begin{aligned} \boldsymbol{u}_{k+1} := \arg\min_{\boldsymbol{u}} \quad & \phi_k^r(\boldsymbol{u}), \\ \text{s.t.} \quad & \boldsymbol{g}_k^r(\boldsymbol{u}) \leq \boldsymbol{0}, \end{aligned} \tag{43}$$

where the modified functions are defined as:

$$\phi_k^r(\boldsymbol{u}) := \phi(\boldsymbol{u}) + (\widehat{\boldsymbol{\lambda}}_k^{\phi,r})^\mathsf{T}(\boldsymbol{y}_r(\boldsymbol{u}) - \boldsymbol{r}_k), \tag{44}$$

$$\boldsymbol{g}_k^r(\boldsymbol{u}) := \boldsymbol{g}_j(\boldsymbol{u}) + \hat{\boldsymbol{\varepsilon}}_k^g + (\widehat{\boldsymbol{\lambda}}_k^{g,r})^\mathsf{T}(\boldsymbol{y}_r(\boldsymbol{u}) - \boldsymbol{r}_k), \tag{45}$$

with the set points for the plant being the corresponding value predicted by the model, $\boldsymbol{r}_k = \boldsymbol{y}_r(\boldsymbol{u}_k)$. The estimated zeroth order modifier, $\boldsymbol{\varepsilon}_k^g$, can be estimated using the transient estimate of the plant constraints:

$$\hat{\boldsymbol{\varepsilon}}_k^g := \hat{\boldsymbol{\mathcal{G}}}_p(\boldsymbol{r}_k) - \boldsymbol{g}(\boldsymbol{u}_k), \tag{46}$$

where $\hat{\boldsymbol{\mathcal{G}}}_p(\boldsymbol{r}_k)$ is the estimate of the steady-state constraints, estimated as the current measured transient value, $\hat{\boldsymbol{\mathcal{G}}}_p(\boldsymbol{r}_k) = \boldsymbol{G}_p(\boldsymbol{u}_p(t), \boldsymbol{y}_p(t))$. The estimated linear modifiers, $\widehat{\boldsymbol{\lambda}}^r$, are calculated with respect to the plant set points, therefore the gradient of the model w.r.t. the set points $\boldsymbol{r}$ is required. This can be achieved by multiplying the gradient by the inverse of the mapping between the respective outputs and the model

inputs:

$$\widehat{\boldsymbol{\lambda}}_k^{\phi,r} \coloneqq \widehat{\nabla}_{\boldsymbol{r}} \varphi_p - \nabla_{\boldsymbol{u}} \phi \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} \right)^+, \tag{47}$$

$$\widehat{\boldsymbol{\lambda}}_k^{g,r} \coloneqq \widehat{\nabla}_{\boldsymbol{r}} \boldsymbol{\mathcal{G}}_p - \nabla_{\boldsymbol{u}} \boldsymbol{g} \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} \right)^+. \tag{48}$$

The gradients of the plant must be estimated using the transient measurements of the plant, therefore steady-state gradient estimation methods cannot be used. Also, the gradients of the plant must be estimated w.r.t the controller set points, rather than the open-loop input variables. This degree of freedom difference poses an issue with using model-based gradient estimation methods due to the unknown controller, which remaps the gradients of the open-loop plant which cannot be identified only using the model. This is analyzed in depth in Section 5.2, and an approach to using the model to estimate the controlled plant gradients using transient measurements and some limited knowledge of how the controller set points and open-loop plant inputs are related is proposed.

4.2. Method UU - Transient

Similar to the original UU method, this method solves both the optimization problem and the modifier in terms of the model inputs $\boldsymbol{u}$.

$$\begin{aligned} \boldsymbol{u}_{k+1} \coloneqq \arg\min_{\boldsymbol{u}} \quad & \phi_k^u(\boldsymbol{u}), \\ \text{s.t.} \quad & \boldsymbol{g}_k^u(\boldsymbol{u}) \leq \boldsymbol{0}, \end{aligned} \tag{49}$$

where the modified functions are defined as:

$$\phi_k^u(\boldsymbol{u}) \coloneqq \phi(\boldsymbol{u}) + (\widehat{\boldsymbol{\lambda}}_k^{\phi,u})^\mathsf{T}(\boldsymbol{u} - \boldsymbol{u}_k), \tag{50}$$

$$\boldsymbol{g}_k^u(\boldsymbol{u}) \coloneqq \boldsymbol{g}(\boldsymbol{u}) + \hat{\boldsymbol{\varepsilon}}_k^g + (\widehat{\boldsymbol{\lambda}}_k^{g,u})^\mathsf{T}(\boldsymbol{u} - \boldsymbol{u}_k). \tag{51}$$

The estimated modifiers are derived in terms of $\boldsymbol{u}$:

$$\widehat{\boldsymbol{\lambda}}_k^{\phi,u} \coloneqq \widehat{\nabla}_{\boldsymbol{r}} \varphi_p \frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} - \nabla_{\boldsymbol{u}} \phi \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} \right)^+ \frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}, \tag{52}$$

$$\widehat{\boldsymbol{\lambda}}_k^{g,u} \coloneqq \widehat{\nabla}_{\boldsymbol{r}} \boldsymbol{\mathcal{G}}_p \frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} - \nabla_{\boldsymbol{u}} \boldsymbol{g} \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} \right)^+ \frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}. \tag{53}$$

Similarly to the transient UR method, the transient measurements are used to calculate estimate the modifiers, however the same issues are present with the transient gradient using model-based techniques.

4.3. Method RR - Transient

In the same way as the steady-state method, this approach formulates a convex approximation of the mapping between the model cost and constraint functions and the model outputs $\boldsymbol{y}_r$. This mapping allows for the direct use of the model w.r.t. the plant control variables. The optimization problem can be defined as:

$$\begin{aligned} \boldsymbol{r}_{k+1} \coloneqq \arg\min_{\boldsymbol{r}} \quad & \varphi_k^r(\boldsymbol{r}), \\ \text{s.t.} \quad & \boldsymbol{\mathcal{G}}_k^r(\boldsymbol{r}) \leq \boldsymbol{0}, \end{aligned} \tag{54}$$

where the modified functions are defined as:

$$\varphi_k^r(\boldsymbol{r}) \coloneqq \varphi^c(\boldsymbol{r}) + (\widehat{\boldsymbol{\lambda}}_k^{\phi})^\mathsf{T}(\boldsymbol{r} - \boldsymbol{r}_k), \tag{55}$$

$$\boldsymbol{\mathcal{G}}_k^r(\boldsymbol{r}) \coloneqq \boldsymbol{\mathcal{G}}^c(\boldsymbol{r}) + \varepsilon_k^{\boldsymbol{g}} + (\widehat{\boldsymbol{\lambda}}_k^g)^\mathsf{T}(\boldsymbol{r} - \boldsymbol{r}_k), \tag{56}$$

where ϕ^c and g_j^c are the convex mappings, with the linear modifiers defined as:

$$\widehat{\boldsymbol{\lambda}}_k^{\phi} \coloneqq \widehat{\nabla}_{\boldsymbol{r}} \varphi_p - \nabla_{\boldsymbol{r}} \varphi^c, \tag{57}$$

$$\widehat{\boldsymbol{\lambda}}_k^{g} \coloneqq \widehat{\nabla}_{\boldsymbol{r}} \boldsymbol{\mathcal{G}}_p - \nabla_{\boldsymbol{r}} \boldsymbol{\mathcal{G}}^c. \tag{58}$$

Similarly to the previous methods, the transient measurements are used to calculate estimate the modifiers. With method RR, the model is replaced by a convex approximation, which cannot be implemented into a model-based gradient estimation method such as NE. This is due to the approximation having no model parameters or outputs, therefore the nominal model must be used instead.

4.4. KKT Matching upon Convergence

In this section, a proposition of the KKT matching upon convergence under certain assumptions is given followed by a proof for all the methods outlined above.

Theorem 1. *Consider the controlled plant optimization problem of* (17)*. The plant operates in closed-loop with an unknown controller where only an open-loop model is available. Assume that, to solve this problem, any of the model-based optimization approaches UR, UU, or RR of Equations* (43)*,* (49) *and* (54) *respectively, are applied using transient measurements, with no noise and perfect gradient estimates at steady-state, i.e.* $\widehat{\nabla}_{\boldsymbol{r}} \varphi_p = \nabla_{\boldsymbol{r}} \varphi_p$*. Then, upon convergence, the converged set points of the plant controller are a KKT point of the plant optimization problem.*

Proof. If any of the approaches UR, UU or RR, outlined in Sections 4.1 to 4.3 respectively, are applied and have converged to a point $\boldsymbol{u}_\infty \coloneqq \lim_{k\to\infty} \boldsymbol{u}_k$ or $\boldsymbol{r}_\infty \coloneqq \lim_{k\to\infty} \boldsymbol{r}_k$. Then the plant must be at steady-state at the set point $\boldsymbol{r}_\infty$ ($\coloneqq \boldsymbol{y}_r(\boldsymbol{u}_\infty)$). As the plant is at steady-state, the use of the transient measurement as an estimate of the steady-state is valid, i.e. $\hat{\boldsymbol{y}}_p = \boldsymbol{y}_p(t) = \boldsymbol{y}_p$, and the estimates of the gradients are also assumed to be correct. With this, the proof can follow the same line of reasoning as the respective steady-state method proof given in François et al. (2016). Hence, $\boldsymbol{r}_\infty$ is a KKT point of the plant. □

It is important to note that in order for these proofs to be valid, knowledge of the controller is not explicitly required, however this is only possible if accurate plant gradients w.r.t. the set points are available. This is challenging, per se, but is made more complicated here because of the mismatch between the degrees of freedom of the plant and model and even worse by the use of model based gradient estimation techniques. This assumption is investigated in Section 5.2, concluding that in order to obtain accurate plant gradient estimates with NE, additional controller information may be required. Another assumption is that the approach has converged, which cannot be guaranteed. This is of particular importance with methods using transient measurements which can, if not adequately accounted for, result in oscillations close to the optimum of the plant.

The resulting approach which is outlined above allows for the rapid optimization of a process, where the plant and model do not have the same degrees of freedom. Thus if the model cannot be easily remodeled to match the degrees of freedom in the plant, optimization via transient MA can still be applied.

5. Plant Gradients vs Model Gradients

This section will analyze two aspects of the gradients involved in the above algorithms. Section 5.1 looks into the first of these aspects of why the use of the pseudoinverse is sufficient to guarantee convergence to the plant optimum *when accurate plant gradients are available.* Section 5.2 looks into the second aspect, relating to the issue regarding the use of an open-loop model in the estimation of the gradient of a controlled plant.

5.1. Reformulating the Model Gradients

This section will look into how the open-loop model gradient can be used to optimize a controlled plant and discuss how the use of a pseudoinverse is appropriate for reformulating the model gradients when *the controlled plant gradients are available*. The objective of the optimization of the controlled plant is to find the controller set points, $\boldsymbol{r}$, which minimize the cost of the plant, whilst satisfying the constraints. In contrast to this, the model-based optimization problem finds the open-loop inputs, $\boldsymbol{u}$, which minimizes the cost of the model, whilst satisfying the constraints. This difference in the degrees of freedom can be overcome by estimating the optimal set points of the plant via the outputs of the model which correspond to the set points of the plant, $\boldsymbol{r}_k = \boldsymbol{y}_r(\boldsymbol{u}_k)$.

For the first-order modifiers, $\boldsymbol{\lambda}$, the gradients of the plant and model must be w.r.t. the same degrees of freedom. Method RR solves this by directly incorporating the degree of freedom change into the model, resulting in a greater plant-model mismatch, but with no degree of freedom issues. Alternatively, methods UR and UU overcome this by reformulating the model-based gradients to be w.r.t. the model outputs which correspond to the plant set points. This gradient transformation required can be derived as follows, starting by the chain rule of the gradient of the model, (assuming row representation of the Jacobian):

$$\frac{\partial \phi}{\partial \boldsymbol{u}} = \frac{\partial \varphi}{\partial \boldsymbol{r}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}. \tag{59}$$

As the controller set points are equal to the model outputs, the above derivative can be simplified by substituting the following:

$$\boldsymbol{r} = \boldsymbol{y}_r(\boldsymbol{u}) \implies \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} = \frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}. \tag{60}$$

Therefore, Equation (59) can be written as:

$$\frac{\partial \phi}{\partial \boldsymbol{u}} = \frac{\partial \varphi}{\partial \boldsymbol{r}} \frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}. \tag{61}$$

From here, the concept of the generalized inverse can be overviewed for taking the inverse of a non-square matrix Ben-Israel et al. (2003). The generalized reflective inverse, labeled with $(.)^g$, is a matrix which satisfies the following conditions:

$$\boldsymbol{A}\boldsymbol{A}^g\boldsymbol{A} = \boldsymbol{A}, \tag{62}$$

$$\boldsymbol{A}^g\boldsymbol{A}\boldsymbol{A}^g = \boldsymbol{A}^g. \tag{63}$$

In addition to these properties, if $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ has a rank of m, then the following is also satisfied:

$$\boldsymbol{A}\boldsymbol{A}^g = \boldsymbol{I}_m. \tag{64}$$

As the set points are determined by the model inputs (via $\boldsymbol{r} = \boldsymbol{y}_r(\boldsymbol{u})$), the number of independent degrees of freedom of the plant set points must be less than or equal to those of the open-loop model inputs, Equation (61) can be right multiplied by the inverse of $\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}}$:

$$\frac{\partial \phi}{\partial \boldsymbol{u}} \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} \right)^g = \frac{\partial \phi}{\partial \boldsymbol{r}} \frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} \right)^g, \tag{65}$$

$$\implies \frac{\partial \phi}{\partial \boldsymbol{r}} = \frac{\partial \phi}{\partial \boldsymbol{u}} \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}} \right)^g. \tag{66}$$

In the case of a change in number of degrees of freedom between the plant and model, $\partial \boldsymbol{y}_r / \partial \boldsymbol{u}$ is non-square, therefore the generalized inverse is not unique. This degree of freedom discrepancy can be overcome by assuming further criteria on the inverse. The Moore-Penrose pseudoinverse overcomes this by enforcing the following:

$$(\boldsymbol{A}\boldsymbol{A}^g)^\mathsf{T} = \boldsymbol{A}\boldsymbol{A}^g, \tag{67}$$

$$(\boldsymbol{A}^g\boldsymbol{A})^\mathsf{T} = \boldsymbol{A}^g\boldsymbol{A}. \tag{68}$$

The pseudoinverse is unique and defined for all matrices. The identities for the generalized inverse, given by Equations (62) and (63), (which also hold for the pseudoinverse) are utilized in the proof of KKT matching of the algorithms (François et al., 2016). Assuming the generalized inverse is the pseudoinverse introduces further plant-model mismatch, however as long as accurate plant gradient estimates are available, the MA scheme can overcome this mismatch.

5.2. Reformulating Model-Based Gradient Estimates of the Plant

In the case that the plant and model have a difference in the number of degrees of freedom, the required gradients of the plant need to be w.r.t. the set points of the controller, as illustrated by Figure 4. When using the transient measurements to speed up convergence of the RTO scheme, estimates of the plant gradient via finite differences is not available. One approach to overcome this is to use the measurements to relate the difference between the plant and model gradients via the model parameters, as in NE. This approach does not solve the degree of freedom difference, resulting in estimates of the open-loop plant. With remodeling the controller, this degree of freedom difference can be overcome, but can be a costly and time consuming task. The following is proposed on how these open-loop estimates can be reformulated to be w.r.t. the set points of the plant without the need of creating a detailed model of the discrepancy.

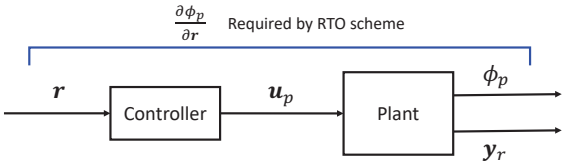


Figure 4: Illustration of the plant gradient estimate required by the proposed RTO scheme.

Proposition 1. *Consider the optimization of a plant which is controlled via an unknown closed-loop controller, given by* (17)*, or more generally a case where the model and plant have a different number of degrees of freedom. In order to solve this problem via the algorithms discussed in Section 4, estimates of the plant gradients w.r.t. the controller set points are required. If matrices* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are defined as follows:*

$$\begin{aligned}\boldsymbol{A}_\phi =& \nabla_{\boldsymbol{u}}\varphi(\boldsymbol{u}_0,\boldsymbol{\theta}) + \nabla^2_{\boldsymbol{u\theta}}\varphi(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+\delta\boldsymbol{y}_p + \\ & (\nabla^2_{\boldsymbol{uu}}\varphi - \nabla^2_{\boldsymbol{u\theta}}\varphi(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+\nabla_{\boldsymbol{u}}\boldsymbol{H})\delta\boldsymbol{u}_p, \qquad (69)\\ \boldsymbol{B} =& \nabla_{\boldsymbol{u}}\boldsymbol{y}_r(\boldsymbol{u}_0,\boldsymbol{\theta}) + \nabla^2_{\boldsymbol{u\theta}}\boldsymbol{y}_r(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+\delta\boldsymbol{y}_p + \\ & (\nabla^2_{\boldsymbol{uu}}\boldsymbol{y}_r - \nabla^2_{\boldsymbol{u\theta}}\boldsymbol{y}_r(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+\nabla_{\boldsymbol{u}}\boldsymbol{H})\delta\boldsymbol{u}_p, \qquad (70)\end{aligned}$$

where $\delta\boldsymbol{y}_p = \hat{\boldsymbol{y}}_p - \boldsymbol{y}_p(\boldsymbol{u}_k^*)$ *and* $\delta\boldsymbol{u}_p = \boldsymbol{u}_p - \boldsymbol{u}_k^*$ *are the differences between the plant and model outputs and inputs respectively. Then, if there is no set point offset, the gradient of the controlled plant*[2] *can be estimated via:*

$$\widehat{\frac{\partial \varphi_p}{\partial \boldsymbol{r}}} = \boldsymbol{A}_\phi(\boldsymbol{B})^g, \qquad (71)$$

where $(.)^g$ *is the generalized reflective inverse.*

Proof. Beginning with matrix $\boldsymbol{A}_\phi$, it can seen that $\boldsymbol{A}_\phi$ can be rearranged into the following:

$$\begin{aligned}\boldsymbol{A}_\phi =& \nabla_{\boldsymbol{u}}\varphi(\boldsymbol{u}_0,\boldsymbol{\theta}) + \nabla^2_{\boldsymbol{uu}}\varphi\delta\boldsymbol{u}_p + \\ & \nabla^2_{\boldsymbol{u\theta}}\varphi(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+(\delta\boldsymbol{y}_p - \nabla_{\boldsymbol{u}}\boldsymbol{H}\delta\boldsymbol{u}_p). \qquad (72)\end{aligned}$$

[2]This proposition is written in terms of cost gradients, but it can be easily generalized to the gradients of the constraints.

Noticing that $\delta\boldsymbol{\theta} = (\nabla_\theta \boldsymbol{H})^+(\delta \boldsymbol{y}_p - \nabla_{\boldsymbol{u}} \boldsymbol{H}\delta \boldsymbol{u}_p)$ leads to:

$$\boldsymbol{A}_\phi = \nabla_{\boldsymbol{u}}\varphi(\boldsymbol{u}_0, \boldsymbol{\theta}) + \nabla^2_{\boldsymbol{uu}}\varphi\delta\boldsymbol{u}_p + \nabla^2_{\boldsymbol{u\theta}}\varphi\delta\boldsymbol{\theta}. \tag{73}$$

It can be seen that this equation is the linearization of the cost of the model, w.r.t. the open-loop inputs and the model parameters. Therefore, $\boldsymbol{A}_\phi$ is an estimate of the open-loop plant cost gradient:

$$\boldsymbol{A}_\phi = \widehat{\frac{\partial \phi_p}{\partial \boldsymbol{u}_p}}, \tag{74}$$

where $\partial\phi_p/\partial\boldsymbol{u}_p$ refers to $\partial\phi_p/\partial\boldsymbol{u}$ at $\boldsymbol{u}_p$. Along the same line of reasoning, the matrix $\boldsymbol{B}$ can be rearranged into the following:

$$\boldsymbol{B} = \nabla_{\boldsymbol{u}}\boldsymbol{y}_r(\boldsymbol{u}_0, \boldsymbol{\theta}) + \nabla^2_{\boldsymbol{uu}}\boldsymbol{y}_r\delta\boldsymbol{u}_p + \nabla^2_{\boldsymbol{u\theta}}\boldsymbol{y}_r\delta\boldsymbol{\theta}. \tag{75}$$

Therefore, $\boldsymbol{B}$ can be interpreted as an estimate of the of the open-loop gradient of the plant outputs,

$$\boldsymbol{B} = \widehat{\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}}. \tag{76}$$

With the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ proven to be estimates of the open-loop gradients of the plant cost and outputs respectively, substituting into the proposed formula for estimating the gradient of the closed-loop plant, the following is obtained:

$$\widehat{\frac{\partial \varphi_p}{\partial \boldsymbol{r}}} = \widehat{\frac{\partial \phi_p}{\partial \boldsymbol{u}_p}} \left(\widehat{\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}} \right)^g. \tag{77}$$

As there is no controller offset, $\boldsymbol{r} = \boldsymbol{y}_r$, therefore with an appropriate choice of generalized inverse, the proposed formula does provide an estimate of the controlled plant gradient. □

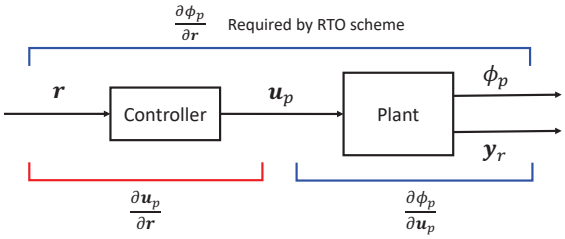


Figure 5: Illustration of the plant gradient estimate required by the proposed RTO scheme, versus the proposed model-based estimation method

The proposed estimate is illustrated in Figure 5. By comparison with Equation (77), it can seen that the controller gradient must be equal to *one* of the generalized inverses of the plant, i.e. only one of the generalized inverses of the open-loop plant is correct:

$$\left(\widehat{\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}} \right)^g = \frac{\partial \boldsymbol{u}_p}{\partial \boldsymbol{r}}. \tag{78}$$

The same procedure can be applied to the constraints, with the $\boldsymbol{A}$ matrix being estimated via:

$$\begin{aligned} \boldsymbol{A}_g =& \nabla_{\boldsymbol{u}}\mathcal{G}(\boldsymbol{u}_0, \boldsymbol{\theta}) + \nabla^2_{\boldsymbol{u\theta}}\mathcal{G}(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+\delta\boldsymbol{y}_p + \\ & (\nabla^2_{\boldsymbol{uu}}\mathcal{G} - \nabla^2_{\boldsymbol{u\theta}}\mathcal{G}(\nabla_{\boldsymbol{\theta}}\boldsymbol{H})^+\nabla_{\boldsymbol{u}}\boldsymbol{H})\delta\boldsymbol{u}_p. \end{aligned} \tag{79}$$

The accuracy of these estimates primarily depends on two factors. The first being the validity of using NE to estimate the open-loop plant cost, constraints and outputs, whilst the second is the accuracy of the choice of generalized reflective inverse of $\boldsymbol{B}$ for estimating the gradient of the controller. This second factor depends on if the number of degrees of freedom between the plant and model changes or not. The following sections investigate both cases.

5.2.1. Special Case - Full Model

In the special case where a full model of the process is available, the open-loop model will have a set of inputs, $\boldsymbol{u}$, and outputs, $\boldsymbol{y}_r$, with these outputs directly corresponding to the controller set points $\boldsymbol{r}$. In the instance where there is no controller offset, the controller differential is unique and equal to the inverse of the plant:

$$\frac{\partial \boldsymbol{u}_p}{\partial \boldsymbol{r}} = \left(\widehat{\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}} \right)^{-1} . \tag{80}$$

As there is no loss of degrees of freedom, this inverse is valid, and the gradient estimate of the controlled plant gradients only depends on the validity of using the model-based gradient estimate technique to estimate the plant gradients.

5.2.2. General Case - Incomplete Model

In the general case, the number of degrees of freedom between the controlled plant and open-loop model may be inconsistent. A loss in degrees of freedom indicates decision information of the plant is not available to the model, i.e. the plant is making decisions for the open-loop inputs which cannot be influenced by the set points. This occurs in such a case where a full model of the process is unavailable or if an unknown control system reduces the number of degrees of freedom[3]. In this case the generalized inverse of the plant is not unique, leading to a choice in how to rectify the degree of freedom difference. This issue was solved for the model-based gradient estimate using the Moore-Penrose pseudoinverse. This solution led to errors in the gradient estimation, however these were inconsequential as the MA scheme can overcome plant-model mismatch. In the case of the plant gradients, erroneous estimates are detrimental to the KKT convergence of the scheme.

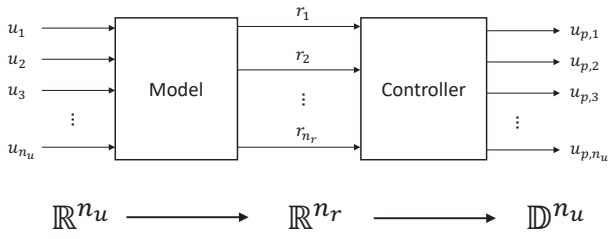


Figure 6: Illustration of the open-loop plant subdomain $\mathbb{D}$

Figure 6 illustrates the transition of the vector spaces from the model inputs $\boldsymbol{u} \in \mathbb{R}^{n_u}$, to the plant set points $\boldsymbol{r} \in \mathbb{R}^{n_r}$, to the open-loop plant inputs $\boldsymbol{u}_p \in \mathbb{D}^{n_u}$. The number of degrees of freedom from one mapping to another cannot increase, therefore if the number of set points to the plant controller is less than the number of inputs to the model, the mapping of the controller set points back to the open-loop inputs space has a lower dimensionality than the model.

An example of this is shown in Figure 7, with a model the two inputs, $\boldsymbol{u} = [u_1, u_2]$, and a controller with one set point, $r = r_1$. The transforms from each vector space is illustrated, and shows the formation of the set point mapping into the open-loop input space, given by $\mathbb{D}$, as a 1-dimensional subdomain in the 2-dimensional open-loop input space, i.e. a line rather than a plane. Finally, as the aim is to find the local gradient of the plant domain w.r.t. the set points; using a linear subspace which is parallel to the plant subdomain at the point of interest will produce the same gradient information as is required for Equation (78). An illustration of the generation of this subspace from the plant input domain is given in Figure 8.

[3]An example is when the ratio between two plant inputs that are also inputs of the model, is kept constant by means of an unmodeled ratio controller.

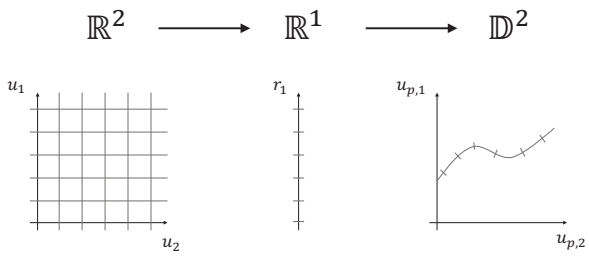


Figure 7: Illustration of the plant domain for an example of two degrees of freedom to one degree of freedom.

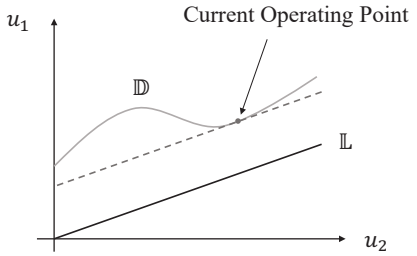


Figure 8: Illustration of the plant subspace from the plant domain in Figure 7

In other words, $\mathbb{D}$ represents where the open-loop plant inputs can exist in the input space, which is dimensionally reduced by the controller. $\mathbb{L}$ represents a subspace which has the same direction in the input space as $\mathbb{D}$ at the current operating point. This subspace can be used to accurately estimate how the open-loop inputs to the plant change with a change in the controller set points.

With the tangential plant subspace defined, the following is proposed on how to estimate the unknown controller gradient using projections from the model input space onto this linear subspace is given. This controller gradient is required for the estimation of the overall plant gradient, using Equations (71) and (78).

Proposition 2. *Consider the optimization of a plant which is controlled via an unknown closed-loop controller (given by* (17)*) which reduces the number of degrees of freedom of the process from n_u to n_r, with no controller set point offset. Denote $\mathbb{L}$ the n_r-dimensional subspace that is tangential to the open-loop plant input domain $\mathbb{D}$, at the current operating point, and $\boldsymbol{P}_{\mathbb{L}}$ the projection matrix from the model inputs $\mathbb{R}$ on to $\mathbb{L}$. If an estimate $\boldsymbol{P}_{\widehat{\mathbb{L}}}$ of $\boldsymbol{P}_{\mathbb{L}}$ is available, then the controller gradient of the plant can be estimated via the following:*

$$\frac{d\boldsymbol{u}_p}{d\boldsymbol{r}} \approx \left(\widehat{\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}} \boldsymbol{P}_{\widehat{\mathbb{L}}} \right)^+ . \tag{81}$$

Proof. We start with the local linearization of the plant and obtain the following relation between the plant outputs and inputs:

$$\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p} d\boldsymbol{u}_p = d\boldsymbol{y}_r. \tag{82}$$

As there is no controller offset, $d\boldsymbol{y}_r = d\boldsymbol{r}$, the above equation can be restated as:

$$\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p} d\boldsymbol{u}_p = d\boldsymbol{r}. \tag{83}$$

Expanding the input change into its corresponding components, $d\boldsymbol{u}_p = \boldsymbol{u}_{p,1} - \boldsymbol{u}_{p,0}$, where $\boldsymbol{u}_{p,0} \in \mathbb{L}, \boldsymbol{u}_{p,1} \in \mathbb{L}$, results in the following linear equation:

$$\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}(\boldsymbol{u}_{p,1} - \boldsymbol{u}_{p,0}) = d\boldsymbol{r}, \qquad \boldsymbol{u}_{p,0}, \boldsymbol{u}_{p,1} \in \mathbb{L}^{n_r}. \tag{84}$$

A space constrained set of simultaneous equations can be rearranged using the projection of $\boldsymbol{u}_p$ on to $\mathbb{L}$ (Rao and Mitra, 1971). This orthogonal projection, given by $\boldsymbol{P}_{\mathbb{L}}$, transforms the unconstrained vector $\boldsymbol{u}_p \in \mathbb{R}^{n_u}$ such that $\boldsymbol{P}_{\mathbb{L}}\boldsymbol{u}_p \in \mathbb{L}^{n_r}$. Therefore, the above set of equation can be restated in the real domain as:

$$\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}\boldsymbol{P}_{\mathbb{L}}(\boldsymbol{u}_{p,1} - \boldsymbol{u}_{p,0}) = d\boldsymbol{r}, \qquad \boldsymbol{u}_{p,0}, \boldsymbol{u}_{p,1} \in \mathbb{R}. \tag{85}$$

This set of equation can be rearranged for the controller gradient, resulting in:

$$\frac{d\boldsymbol{u}_p}{d\boldsymbol{r}} = \left(\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}\boldsymbol{P}_{\mathbb{L}}\right)^{+}. \tag{86}$$

This is a key result, indicating that the controller gradient is *equal* to the inverse of the plant gradient multiplied by the projection of $\boldsymbol{u}_p$ onto $\mathbb{L}$. As the tangential subspace is unknown, it must be estimated resulting in an estimate of the controller gradient. Therefore, if $\boldsymbol{P}_{\widehat{\mathbb{L}}}$ is an estimate of the projection onto the true tangential subspace of the plant and the plant gradient has been estimated via NE, then:

$$\frac{d\boldsymbol{u}_p}{d\boldsymbol{r}} \approx \left(\widehat{\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p}}\boldsymbol{P}_{\widehat{\mathbb{L}}}\right)^{+}. \tag{87}$$

□

Referring back Proposition 1, Equation (71) can be rewritten as the definite equation:

$$\widehat{\frac{\partial \phi_p}{\partial \boldsymbol{r}}} = \boldsymbol{A}_{\phi}(\boldsymbol{B}\boldsymbol{P}_{\widehat{\mathbb{L}}})^{+}, \tag{88}$$

$$\widehat{\frac{\partial \boldsymbol{g}_p}{\partial \boldsymbol{r}}} = \boldsymbol{A}_{g}(\boldsymbol{B}\boldsymbol{P}_{\widehat{\mathbb{L}}})^{+}. \tag{89}$$

This result allows for the estimation of the controlled plant gradients using model-based gradient estimation via the open-loop model. The estimate of the subspace tangential to the plant at the current operating point can be estimated via several different approaches. One such approach is to use limited extra information to estimate the subspace, such as roughly between which variables the degree of freedom reduction takes place. Alternatively a data-driven approach can be used such as using steady-state data at the model operating point to estimate the subspace before the RTO is undertaken, or via measurements during the RTO.

In the special case of when the number of degrees of freedom is consistent between the plant and model, the plant subspace $\mathbb{L}$ is equal to the model vector space, therefore the projection from the model input space to the plant subspace is equal to the identity matrix, and therefore Equation (81) can be rearranged into Equation (80).

As discussed earlier, the purpose of $\mathbb{L}$ is to more accurately estimate the gradient of the controlled plant. The estimation of this subspace can be thought of as a rudimentary parameter estimator, using measurements (and possibly a predefined structure) to estimate the relationships between the open-loop plant inputs and the controller set points. In order for the approach to maintain the advantages of model based gradient estimation approaches, this estimation of the $\mathbb{L}$ space must be simple and require as few measurements as possible.

It has been assumed that there is no controller offset for the above proposition and proof. This assumption affects the validity of Equation (86), and with an offset in the controller, this is only an approximation of the true controller mapping. However, as estimates are used to approximate this formula, the impact of an offset can just be accounted for as additional errors in the estimate in Equation (87).

5.2.3. Example - Unknown Control System

As an example of calculating the controller gradient using an open-loop model and some limited information of the controller is analyzed on the following example. In this example the open-loop process has 3 inputs, $\boldsymbol{u} = [u_1, u_2, u_3]$, whilst the controlled process has 2 degrees of freedom, $\boldsymbol{r} = [r_1, r_2]$, indicating a loss of 1 degree of freedom. The optimum of the model is given by $\boldsymbol{u}^* = [9, 11, 100]$. Assuming the open-loop process can be described by the following local gradient matrix,

$$\frac{\partial \boldsymbol{y}_r}{\partial \boldsymbol{u}_p} = \begin{bmatrix} -1 & 1 \\ -2 & -1 \\ 1 & 1 \end{bmatrix}^{\mathsf{T}}. \tag{90}$$

Initially looking at the actual plant gradients, assuming the unknown control law is characterized by the non-linear equation:

$$u_2 = u_1^{\alpha}, \tag{91}$$

where α is a constant. Given the true control law of the plant has the constant, $\alpha_p = 1.2$, and that the current operating point is at $\boldsymbol{u}_p = [8.0, 12.1, 90]$; independent of the relationship between the set points and the open-loop control scheme, the true controller gradient can be calculated via Equation (86) as

$$\frac{\partial \boldsymbol{u}_p}{\partial \boldsymbol{r}} = \begin{bmatrix} -0.262 & -0.476 & -0.214 \\ 0.262 & 0.476 & 1.214 \end{bmatrix}^{\mathsf{T}}. \tag{92}$$

Next looking at if some limited knowledge of the control system is known, and how this knowledge can be used to formulate the estimate. The unknown information is in the form of the control law given by Equation (91). It is assumed that the form of the control law is known, but the true constants for the plant are unknown. Using the model optimal inputs, the missing information can be approximated:

$$\alpha = \frac{log(u_2^*)}{log(u_1^*)} \approx 1.09. \tag{93}$$

This assumption allows for the estimate of the plant open-loop input subspace to be defined via the span:

$$\widehat{\mathbb{L}} = span\left(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\right), \tag{94}$$

where $\boldsymbol{v}$ are the vectors of input differentials:

$$\boldsymbol{v}_i = \left[\frac{\partial u_1}{\partial u_i}, \frac{\partial u_2}{\partial u_i}, \frac{\partial u_3}{\partial u_i}\right]^{\mathsf{T}}, \tag{95}$$

resulting in a subspace of,

$$\widehat{\mathbb{L}} = span\left(\begin{bmatrix} 1 \\ 1.334 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.750 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right), \tag{96}$$

resulting in a subspace of two linearly independent vectors of:

$$\widehat{\mathbb{L}} = \left\{ c_1 \begin{bmatrix} 1 \\ 1.334 \\ 0 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \middle| \, c_1, c_2 \in \mathbb{R} \right\}. \tag{97}$$

Where 1.334 is the estimated gradient of u_2 w.r.t. u_1. From this, the projection can be estimated (Dunford, 1988), using the following formula:

$$\boldsymbol{P}_{\mathbb{L}} = \boldsymbol{V}(\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V})^{-1}\boldsymbol{V}^{\mathsf{T}} \tag{98}$$

where $\boldsymbol{V}$ is the matrix of independent vectors in the span of $\mathbb{L}$,

$$\boldsymbol{V} = \begin{bmatrix} 1 & 0 \\ 1.334 & 0 \\ 0 & 1 \end{bmatrix}, \tag{99}$$

thus,

$$\boldsymbol{P}_{\widehat{\mathbb{L}}} = \begin{bmatrix} 0.360 & 0.480 & 0 \\ 0.480 & 0.640 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{100}$$

From the projection, the controller gradient can be estimated via Equation (81), resulting in:

$$\left(\frac{\partial \boldsymbol{y}_p}{\partial \boldsymbol{u}_p}\boldsymbol{P}_{\widehat{\mathbb{L}}}\right)^+ = \begin{bmatrix} -0.300 & -0.400 & -0.100 \\ 0.300 & 0.400 & 1.100 \end{bmatrix}^\mathsf{T}. \tag{101}$$

This estimate is close to the true plant estimate (i.e. comparing (92) to (101)), with a relatively small error in the values and all in the correct direction. In comparison to the direct use of the pseudoinverse:

$$\left(\frac{\partial \boldsymbol{y}_p}{\partial \boldsymbol{u}_p}\right)^+ = \begin{bmatrix} -0.357 & -0.286 & 0.071 \\ 0.571 & -0.143 & 0.286 \end{bmatrix}^\mathsf{T}, \tag{102}$$

it is evident that there is a great improvement in the accuracy of the different methods.

5.2.4. Summary of Method

Algorithm 2 summarizes how transient MA can be applied to the case where the plant and model have a different set of degrees of freedom.

Algorithm 2 MA Algorithm with NE and degree of freedom difference

Begin
Choose approach (UR/UU/RR)
Initialize all modifiers to $\mathbf{0}$
Modify the relevant model-based optimization problem (43), (49) or (54)
for $k = 0 \rightarrow \infty$
1: Solve modified model-based optimization problem and deduce the set points to apply to the plant
2: Implement set points on plant
3: Take measurements of the plant at specified time step
4: Estimate $\widehat{\mathbb{L}}$ and $\boldsymbol{P}_{\widehat{\mathbb{L}}}$
5: Estimate plant gradients (88) and (89)
6: Calculate the modifiers
7: Apply exponential filter
8: Reformulate modified model-based problem
end

The resulting approach of using projections to rectify some of the missing information in the gradient estimation is objectively less effective than a complete remodel of the process to obtain a model with match degrees of freedom to the plant. However, it enables the application of MA with transient measurements without remodeling, even if model-based gradient estimation is used. This is made possible via the inclusion of minor additional information in the gradient estimation, and the gains from the inclusion of this additional information results in rapid and potentially significant improvements to the operating conditions, as illustrated in the following case studies.

6. Case Study 1 - Williams-Otto Reactor

As an initial example, a case study of a simple CSTR based on the Williams-Otto reactor (Williams and Otto, 1960) with structural plant-model mismatch is illustrated. The plant has a control scheme with additional restrictions on the open-loop input variables reducing the number of degrees of freedom. This control scheme can be remodeled to rectify this loss of degrees of freedom, however as previously discussed this is a costly option and is not necessary. The plant and model are discussed briefly below with a detailed description in the supplementary material, along with a description of the control scheme, followed by the results of the new MA scheme.

6.1. Open-loop Plant

The simulated plant follows a 3-reaction system, with two reactants (A and B), an intermediate component (C), two products (P and E), and a waste component (G). This system is described by the following set of reactions:

$$\begin{aligned} A + B &\xrightarrow{k_1} C, \\ B + C &\xrightarrow{k_2} P + E, \\ C + P &\xrightarrow{k_3} G. \end{aligned}$$

The reaction rate constants, k_j, can be calculated using the Arrhenius equation, and it is assumed that the CSTR is isothermal at the chosen temperature with ideal mixing. A detailed description of the plant is given in the supplementary material. The open-loop inputs are the mass flow rates of components A and B in $\mathrm{kg\,s^{-1}}$, given by F_A and F_B respectively, and the reactor temperature in Kelvin, given by T, thus $\boldsymbol{u} = [F_A, F_B, T]$. The objective function and constraints on the system can be written as:

$$\begin{aligned} \Phi(\boldsymbol{u}, \boldsymbol{y}) &= 76.23F_A + 114.34F_B \\ &\quad - 1143.38X_P F - 25.92X_E F, \qquad (103) \\ G(\boldsymbol{u}, \boldsymbol{y}) &= X_G - 0.08 \leq 0, \qquad (104) \end{aligned}$$

where X_i is the mass fraction of component i in the reactor and F is the total reactor inlet flow rate. The objective function relates to the degree of cost per unit time, with the objective of the optimization scheme being to minimize this, whilst satisfying the system constraints.

6.2. Open-loop Model

The model of this process, based on the model in (Roberts, 1979), is a steady-state model with the same inputs as the *open-loop* plant. This model has a significant degree of mismatch to the plant via a simplified, two reaction mechanism. This system does not contain the intermediate component C, and can be described as follows:

$$\begin{aligned} A + 2B &\xrightarrow{k_1^*} P + E, \\ A + B + P &\xrightarrow{k_2^*} G, \end{aligned}$$

where k_j^* are the model reaction constants, again calculated via the Arrhenius equation $k_j^* = A_j^* e^{-E_j^*/T}$, where A_j^* and E_j^* are the pre-exponential factor and activation energy of reaction j of the model respectively. NE is used for the gradient estimate of the plant, using the activation energies as the model parameters.

Parameter	Value	Unit
A_1^*	2.189×10^8	s^{-1}
A_2^*	4.310×10^{13}	s^{-1}
E_1^*	8075	K
E_2^*	12 400	K

Table 1: Simulated model nominal parameters

6.3. Control Scheme

To illustrate the new method, it has been chosen that an unmodeled flow controller is included in the plant. This ratio control is on the inputs, forcing the flow rate of A and B to fit the ratio given as:

$$F_{A,p} = \frac{F_{B,p}}{\alpha_p}, \tag{105}$$

where $\alpha_p = 2.4$ is the flow ratio. The control scheme uses the total flow rate, $F = F_A + F_B$, as the control variable to determine the individual component inlet flow rates. Additionally, the concentration of component A is controlled via a PI controller on the temperature of the reactor with the following control law:

$$T_p = K_P(X_{A,s} - X_{A,p}) + K_I \int_0^t (X_{A,s} - X_{A,p})dt', \tag{106}$$

where $K_P = -200\text{K}$ and $K_I = -10\text{K}\,\text{s}^{-1}$ are the proportional and integral gains, respectively. Thus the plant degrees of freedom are $\boldsymbol{r} = [F_s, X_{A,p}]$. The settling time of the controlled plant is in the order of 20 minutes, indicating that using steady-state measurements to estimate the gradient, each RTO iteration would take of the order of an hour to complete.

For this case study, limited knowledge of the control scheme is known, using this knowledge the plant input subspace can be estimated, and the projection can be used in Equations (88) and (89). The assumed knowledge is that the flow rate of A and B are linked via a control law, and it is incorrectly assumed that this relation is a power law given by,

$$F_{B,p} = F_{A,p}^{1.5} + \chi, \tag{107}$$

As the measured value of the plant flow rate variables will not necessarily match a fixed power law, the variable χ is used to correct for the discrepancy. Using this control law, the estimate of the subspace can be written as:

$$\widehat{\mathbb{L}} = \left\{ c_1 \begin{bmatrix} 1 \\ \alpha \\ 0 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \middle| \, c_1, c_2 \in \mathbb{R} \right\}. \tag{108}$$

where $\alpha = 1.5F_{A,p}^{0.5}$ is the gradient of between the inputs at the current operating point, estimated using the assumed control law. The orthogonal projection on to $\widehat{\mathbb{L}}$ can be written as:

$$P_{\widehat{\mathbb{L}}} = \begin{bmatrix} \frac{1}{1+\alpha^2} & \frac{\alpha}{1+\alpha^2} & 0 \\ \frac{\alpha}{1+\alpha^2} & \frac{\alpha^2}{1+\alpha^2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{109}$$

using Equation (98).

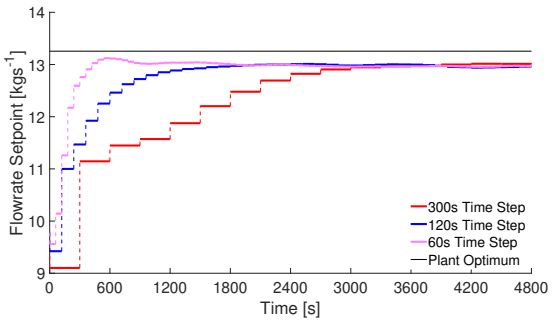


Figure 9: Profile of the flowrate set point for the UR algorithms at different time steps.

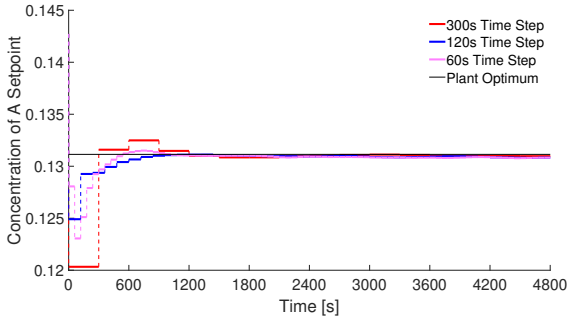


Figure 10: Profile of the concentration of component A set point for the UR algorithm at different time steps.

6.4. Results

The true optimum of the plant is at $\boldsymbol{r}_p^* = [13.2, 0.131]$, with an objective function value of $-\phi_p^* = 210.3$. Whilst the nominal model optimum is at $\boldsymbol{u}^* = [2.9, 6.9, 78.3]$, with a corresponding plant operating point of $\boldsymbol{r}_0 = \boldsymbol{y}_r = [9.8, 0.143]$, resulting an objective function value of $-\phi_p = 151.3$.

Firstly looking at algorithm UR, for different time steps, ranging from 5 minutes down to 1 minute. All three time steps converge to a near optimal point. As shown in Figures 9 and 10, the time for the set points to converge are in the order of a few settling times of the plant, with the lowest time step reaching the converged set point in approximately one plant settling time. As the gradient estimation is imperfect, the converged point is not guaranteed to be the true plant optimum, however for the given model, the use of NE is appropriate and provides a good estimate of the plant gradient. This results in good convergence, as shown by Figures 11 and 12.

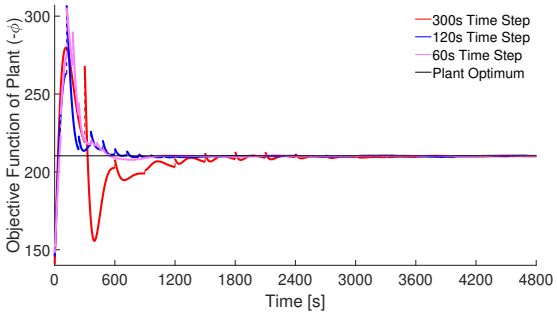


Figure 11: Profile of the objective function for the UR algorithm at different time steps.

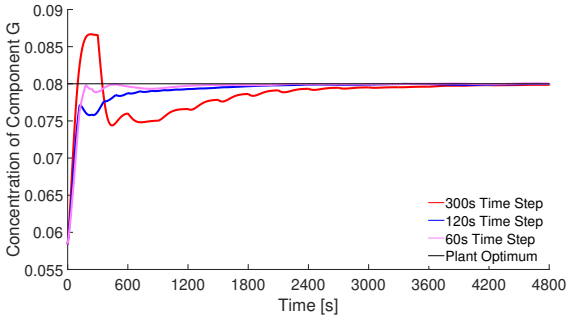


Figure 12: Profile of the concentration of component G (active constraint) for the plant using the UR algorithm at different time steps.

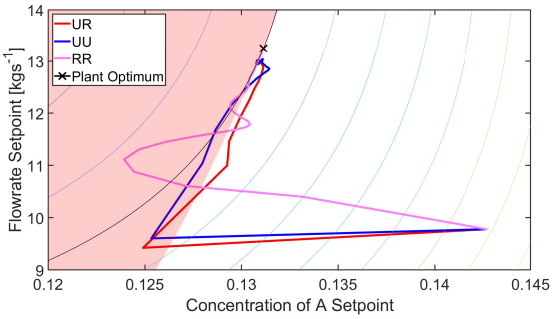


Figure 13: Evolution of the set points for the three different proposed methods. All three methods are simulated with a time step of 120 seconds and a modifier gain of 0.2. The infeasible region of the plant is highlighted in red, whilst the cost contours are shown with an increasing objective function from left to right in steps of 10. The optimal cost contour shown in black with the plant optimum shown as a cross.

Next looking at how different algorithms compare. For the RR method, a convex approximation of the set points predicted by the model and the model cost and constraints is required. This can be achieved by simulating the model at different operating points and fitting a quadratic function to the data (François and Bonvin, 2013), resulting in an approximation in the form of:

$$\phi^c(\boldsymbol{r}) = \phi^* + \boldsymbol{\alpha}(\boldsymbol{r} - \boldsymbol{y}_r^*) + \frac{1}{2}(\boldsymbol{r} - \boldsymbol{y}_r^*)^\mathsf{T}\boldsymbol{Q}(\boldsymbol{r} - \boldsymbol{y}_r^*), \tag{110}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{Q}$ are the quadratic coefficients. $\boldsymbol{Q}$ is chosen to such that the eigenvalues are positive, enforcing the convexity of the approximation.

Figure 13 shows how the set points change with each iteration towards the optimum of the plant. As a note, the profit contours and infeasible region are for when operating the plant at steady-state, therefore despite method RR appearing to go highly infeasible, in reality the concentration of component G remains closer to the constraint line. Also it should be remembered that plant constraints are defined at steady state, while all figures display transient quantities. This is also the reason why the instantaneous plant cost can exceed the optimal steady-state cost before steady state has been reached. Methods UR and UU have the same method of plant gradient estimation, therefore the converged point is consistent between all methods, approaching a near optimal point for the plant. Method RR does not have an iterating model input $\boldsymbol{u}_k^*$, therefore the nominal model optimum is used instead where appropriate, resulting in a different converged point.

Finally, investigating how the model influences the convergence of the method. This can be carried out by changing the parameters of the model and observing the change in the evolution of the set points. This is achieved by modifying the activation energies of the model to match the two case studies in (François

	E_1^* [K]	E_2^* [K]
Case 1	8050	12500
Case 2	8100	12300

Table 2: Activation energies of cases 1 and 2

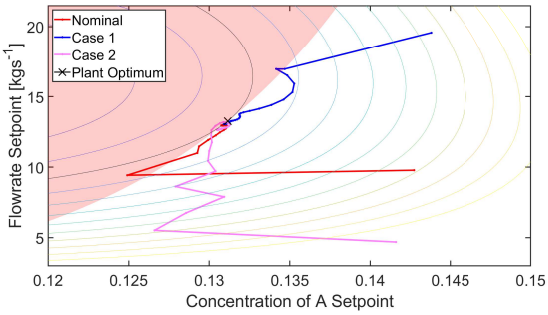


Figure 14: Evolution of the set points for the three different models. All three models are simulated using the UR method with a time step of 120 seconds and a modifier gain of 0.2.

et al., 2016), outlined in Table 2, whilst keeping the other parameters the same as the nominal model. As can be seen from Figure 14, the change in the parameters of the model greatly changes the model optimum. Despite this, all three cases converge to close to the true optimum of the plant.

6.5. Alternative Methods

Alternative methods of using transient measurements in the optimization of a process are compared. The method proposed by Rodríguez-Blanco et al. (2017), as discussed briefly in Section 3.2, uses a measurement based approach to estimating the gradient, and thus can be directly applied to a case which has a differing number of degrees of freedom. A comparison of using NE using controller projections, and RELS is given in the figures below. These simulations have been applied with and without measurement noise considered on the plant output. Firstly, without noise to illustrate the convergence properties of each method.

The RELS approach has been implemented as proposed by Rodríguez-Blanco et al. (2017). The parameters of the simulation are as follows: $\alpha = 0.7$, $\boldsymbol{K} = 0.2$, with a persistent excitation of $a = 0.02$ with a scaled plant set point. The resulting simulation with a transient sampling time of 600s converges to the plant optimum, however is sensitive to the parameter choice.

As can be seen from the Figures 15 and 16, using the RELS approach to gradient estimation provides good convergence properties to the plant optimum, however the fact that RELS takes longer to converge is

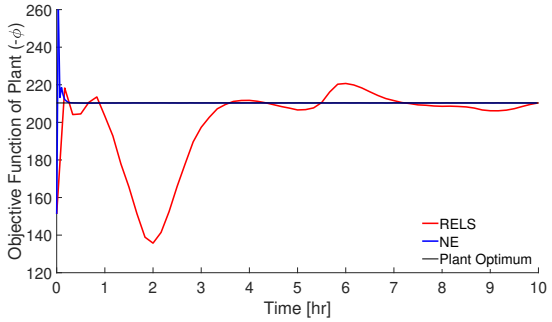


Figure 15: Profile of the measured objective function for different gradient estimation approaches at the end of the RTO iteration. Both approaches utilizing the UR method with NE using 120s time steps and RELS using 600s time steps.

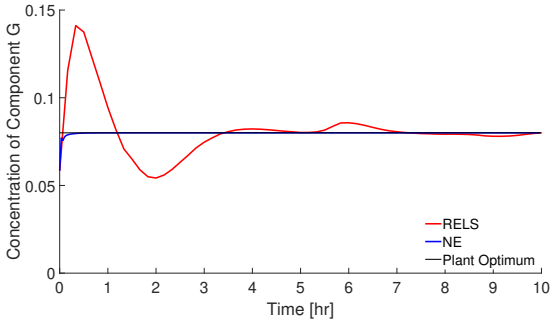


Figure 16: Profile of the measured concentration of component G for different gradient estimation approaches at the end of the RTO iteration.

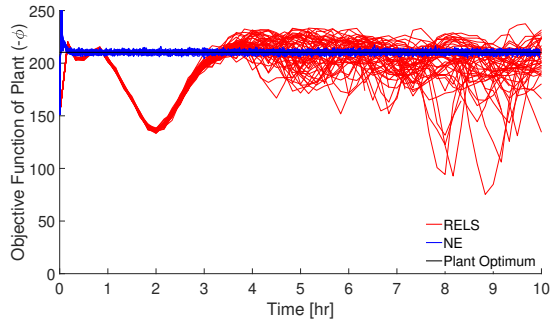


Figure 17: Simulation of 50 instances with measurement noise. Measured cost at each RTO iteration is displayed.

expected since the gradient estimation is purely based on measurements, while NE uses a model which may lead to less accurate estimates, results in faster convergence (François et al., 2012).

It is important to note that the approach proposed by this article uses additional information which is required for estimation of the plant gradients. This information is not required by measurement based gradient approximation techniques, thus definitive conclusions between the approaches cannot be made. The additional information is limited with minimal information of the controller yet results in a near optimal solution. Without the use of this additional information results in poor convergence, with a greater (worse) objective function value compared with the model based optimum.

Next, both approaches are simulated using measurement noise on the outputs of the plant. This noise can be characterized by the following, $\boldsymbol{y}'_p = \boldsymbol{y}_p + X_y$, where $X_y \sim N(0, 0.0001^2)$. This results in a noise in the objective function of approximately $\phi'_p = \phi_p + X_\phi$, where $X_\phi \sim N(0, 0.9^2)$. The proposed approach was applied with a time step of 120s, and the method of using RELS for the gradient estimation has a time step of 600s.

As can be seen in Figures 17 and 18, the noise impact when using the measurement based approach is large, resulting in a poor performance with an average objective function of $-\phi'_p = 192.9$ and consistent violation of the constraints.

7. Case Study 2 - Distillation Column

As a more elaborate example with more complex dynamics to illustrate the proposed method on, a simulation of a distillation column was developed, based on (Diehl et al., 2001). This example is of a high purity binary distillation column, used in the separation of methanol and n-propanol. The column has 40 trays ($N = 40$), with the feed on tray 21 ($N_f = 21$), a partial reboiler, and total condenser. The open-loop system inputs are the reboiler heat duty, Q, and the reflux volumetric flow rate, L_{vol}. The control scheme

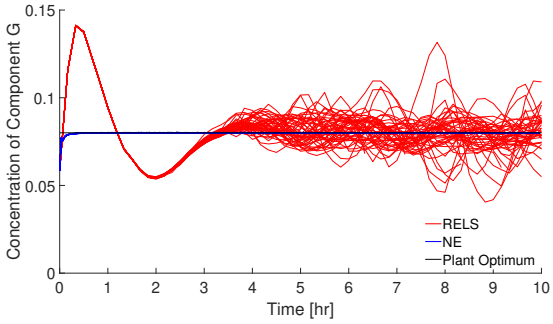


Figure 18: Simulation of 50 instances with measurement noise. Measured cost at each RTO iteration is displayed.

investigated is based around controlling the temperatures on the key stages of trays 14 and 28. A brief description of the plant and model is given below, with a detailed outline of the governing equations in the supplementary materials.

The feed is subject to regular step changes in composition, and flow rate, due to a change in source of the feed stream, resulting is a changing set of optimal operating conditions. Therefore, the use of a transient method will be preferred to find the optimum, or close to, since convergence can be potentially achieved in a single iteration to steady state, avoiding multiple iterations to steady state while the plant optimal conditions have changed. The objective of the optimization is to maximize the operating profit, whilst meeting the specified distillate concentration:

$$\Phi(\boldsymbol{u}, \boldsymbol{y}) \coloneqq 0.1Q - 6.4D, \tag{111}$$

$$G(\boldsymbol{u}, \boldsymbol{y}) \coloneqq 0.99 - x_D, \tag{112}$$

where Q is in kW and D is in kmol h^{-1}. The objective function is based on the cost of electricity (\$0.1/kWh) and product sale price (\$6.4/kmol).

7.1. Open-Loop Plant

The plant has been simulated using an equilibrium-based dynamic algebraic system (DAE). The trays are referred to by $l = 1, 2, ..., N$; the reboiler by $l = 0$; and the condenser by $l = N + 1$. The corresponding temperatures are denoted by $T_0, ..., T_{N+1}$. The composition of methanol in the liquid phase on tray l is denoted by x_l, and as the process is a binary system, the concentration of the n-propanol is denoted with $1 - x_l$ for $l = 0, ..., N + 1$.

A detailed description of the plant with all governing equations is given in the supplementary material. To summarize, this 204 state DAE system is formed of 82 differential states and 122 algebraic states. The differential states are formed of 42 composition states x_l for $l = 0, ..., N+1$, and 40 tray molar holdup states n_l for $l = 1, ..., N$. Whilst the algebraic states are formed of 41 volumetric fluxes V_l for $l = 0, ..., N$, 40 liquid fluxes L_l for $l = 1, ..., N$, and 41 temperatures T_l for $l = 0, ..., N$. The process is based around each tray being an individual unit which is a fixed percent from equilibrium, given by a Murphree plate efficiency.

7.2. Model

An equilibrium based model has been developed which is a simplification of the plant, the simplifications are summarized as follows

1. The molar-hold up of the trays are assumed to be constant
2. The reboiler and condenser molar holdup are assumed to be constant (with variable volumetric holdup)
3. The condenser is assumed to operate at the saturated liquid temperature
4. The model is steady-state

These simplification have the following impact on the model equation. Firstly, the molar holdup differential can be set to 0 for $l = 0, ..., N + 1$:

$$\frac{\partial n_l}{\partial t} = 0. \tag{113}$$

Therefore, the liquid flow rates can be directly calculated via the total mass holdup for the reboiler and trays, $l = 0, ..., N$:

$$L_l = V_{l-1} + L_{l+1} - V_l + F_l. \tag{114}$$

The resulting model has 125 states, with no molar holdup and liquid flux states, but an additional temperature state for the condenser. In addition to the simplifying discussed above, the Murphree plate efficiencies are assumed to be different between the plant and model with efficiencies. The feed is assumed to be fixed with a flow rate of $14\,\mathrm{L\,h^{-1}}$ and a concentration of 0.32 at 71°C. This results in structural mismatch between the open-loop plant and model. The parameters used by NE are the plate efficiencies, feed flow rate and feed composition.

7.3. Control Scheme

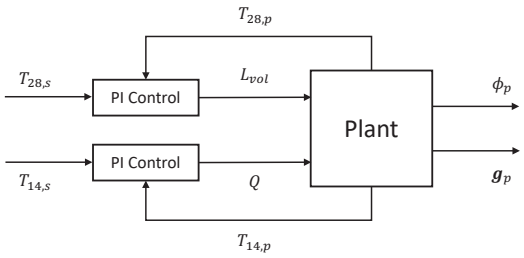


Figure 19: Illustration of the control system for the distillation column

In addition to the open-loop plant model mismatch, there is a difference in the nature of the degrees of freedom between the plant and model. This difference results from the control scheme, which is unknown to the open-loop model and resulting optimization scheme. In this scenario, a decoupled PI control has been implemented to control for the temperatures of trays 14 and 28, $\boldsymbol{r} = [T_{14,s}, T_{28,s}]$. One control loop uses the condenser reflux to control for the temperature of tray 28, whilst the other uses the reboiler duty to control for the temperature of tray 14. The control laws are as follows:

$$L_{vol} = K_{28}^P(T_{28,s} - T_{28,p}) + K_{28}^I \int_0^t (T_{28,s} - T_{28,p})dt', \tag{115}$$

$$Q = K_{14}^P(T_{14,s} - T_{14,p}) + K_{14}^I \int_0^t (T_{14,s} - T_{14,p})dt', \tag{116}$$

with the controller gains given in Table 3. With this control scheme, there is no change in the number of degree of freedom between the plant and model, therefore Equation (80) can be used to estimate the controller gradient from the plant gradient estimate. The settling time of the plant with this control scheme is of the order of several hours.

7.4. Results

The nominal optimization values are outlined in Table 4, assuming the feed begins at the conditions assumed by the model. The optimal controller set points suggested by the model, $\boldsymbol{r}_0 = \boldsymbol{y}_r(\boldsymbol{u}_0)$, are feasible yet suboptimal for the plant. As both the objective function and the constraints are primarily interested in the distillate, the process optimum is more heavily influenced by the temperature on tray 28.

Gain	Value	Units
K_{28}^P	-0.1	$\mathrm{L\,h^{-1}\,K^{-1}}$
K_{28}^I	-0.0002	$\mathrm{L\,h^{-1}\,K^{-1}\,s^{-1}}$
K_{14}^P	0.1	$\mathrm{kW\,K^{-1}}$
K_{14}^I	0.0002	$\mathrm{kW\,K^{-1}\,s}$

Table 3: Controller gains for the decoupled PI control system of the distillation column

	Variable	Value	Units
Plant Optimum	$T_{28,s}^*$	69.89	°C
	$T_{14,s}^*$	83.25	°C
	$-\phi_p^*$	0.1846	
	$x_{D,p}^*$	0.9900	
Model Optimum	L_{vol}^*	4.392	$\mathrm{L\,h^{-1}}$
	Q^*	2.465	kW
Plant Initial Point	$T_{28,s}$	68.25	°C
	$T_{14,s}$	80.91	°C
	$-\phi_p$	0.1718	
	$x_{D,p}$	0.9930	

Table 4: Nominal Results. First set of solutions are the controlled plant optimal solutions, followed by the open-loop model optimal solution, followed by the plant solution operated at the model optimal solution.

Variable	$t < 20$h	$t > 20$h	Units
F	14	13.2	$\mathrm{L\,h^{-1}}$
x_F	0.32	0.34	
T_F	71	71	°C

Table 5: Feed conditions into the column

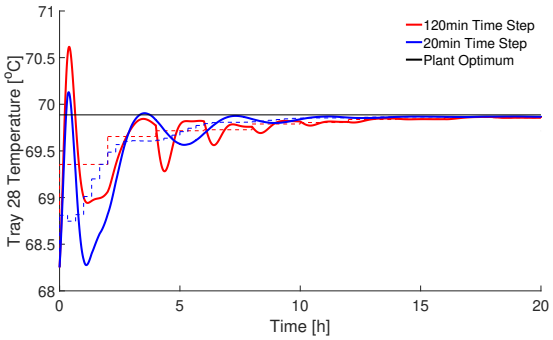


Figure 20: Evolution of the temperature of tray 28 for the first 20 hours of operation using the transient UR method. Solid line represents the actual plant state, whilst the dashed line represents the controller set point.

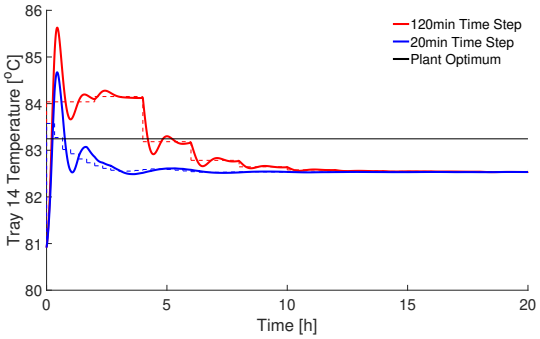


Figure 21: Evolution of the temperature of tray 14 for the first 20 hours of operation using the transient UR method. Solid line represents the actual plant state, whilst the dashed line represents the controller set point.

The scenario investigated in this case study looks at a 40 hour time window, starting from the operating point predicted by model optimum, running the plant at the standard inlet conditions (i.e. those assumed by the model), then after 20 hours a change in the feed concentration and feed flow rate observed over another 20 hours. The feed conditions are summarized in Table 5.

Initially looking at the first 20 hours, Figures 20 and 21 illustrate how the set points to the temperatures evolve over time, with the set point of tray 28 rapidly converging a near optimal state, with a faster convergence for the shorter time step. However, the set point for the temperature of tray 14 converges to a point further away from the true optimum, likely due to the less impact this variable has on the cost and constraints.

Looking at the associated objective function and constraints on Figures 22 and 23, the converged point is very close to the true optimal cost, whilst remaining feasible. The cost is a function of the flow rate of the distillate, whilst the constraint is a function of the composition. Therefore, for a change in the inputs to the plant, the change in the flow rate reacts faster than the composition, thus an overly aggressive scheme (i.e. high value of filter gain) results in a degree of instability. The result is that the convergence time is limited to be in the order of a couple of settling times of the plant.

Next, looking at the following 20 hours with a step in the feed flow rate and concentration. If no action is taken, the plant will converge to an infeasible point, Figures 24 and 25 illustrates the evolution of the set points, while Figures 26 and 27 illustrate the objective function and the constraints.

Again, the set points rapidly approach their converged state with a near optimal convergence of the objective function, with the shorter time step converging faster. Reducing the time step further has the same convergence time as for the larger time steps (i.e. for less than 20 minutes). This is likely due to the slow settling time of the plant being the limiting factor for speeding up the convergence.

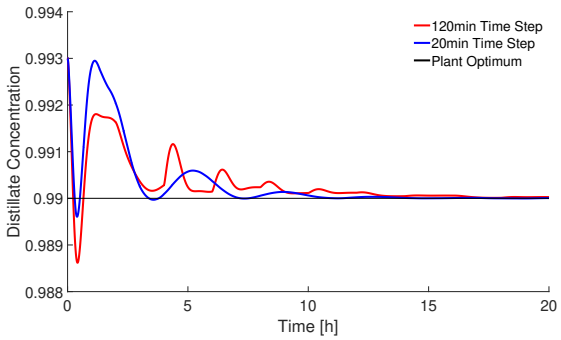


Figure 22: Evolution of the objective function for the first 20 hours of operation using the transient UR method.

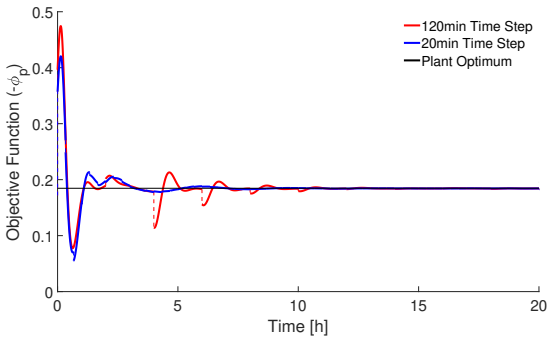


Figure 23: Evolution of the concentration of the distillate (constraint) for the first 20 hours of operation using the transient UR method.

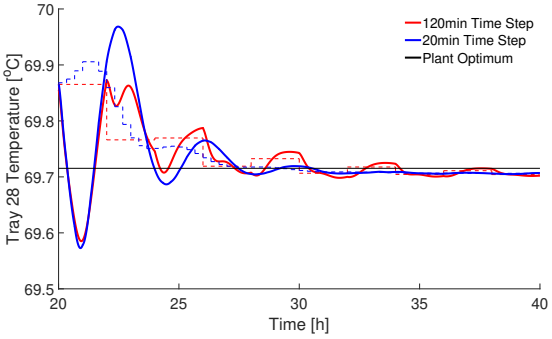


Figure 24: Evolution of the temperature of tray 28 for the following 20 hours of operation using the transient UR method. Solid line represents the actual plant state, whilst the dashed line represents the controller set point.

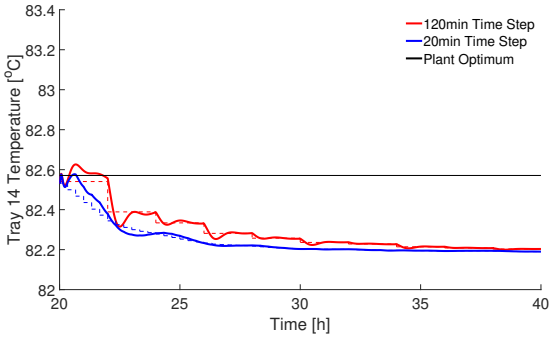


Figure 25: Evolution of the temperature of tray 14 for the following 20 hours of operation using the transient UR method. Solid line represents the actual plant state, whilst the dashed line represents the controller set point.

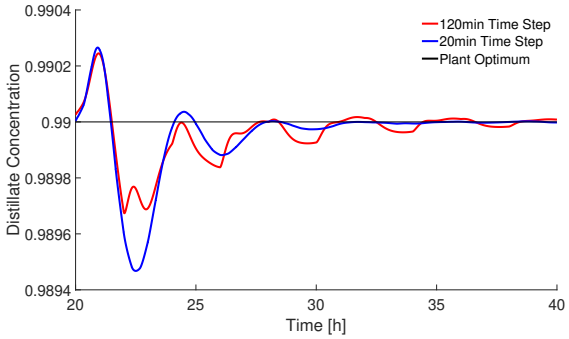


Figure 26: Evolution of the objective function for the following 20 hours of operation using the transient UR method.

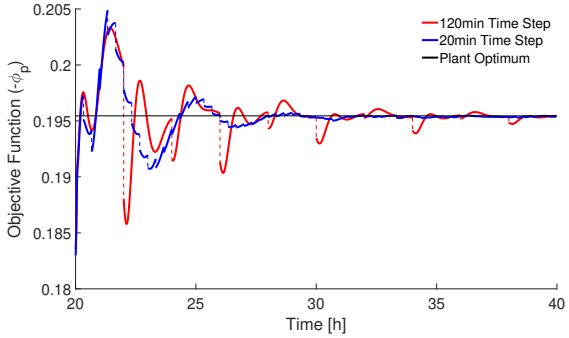


Figure 27: Evolution of the concentration of the distillate (constraint) for the following 20 hours of operation using the transient UR method.

One drawback of using a model-based gradient estimation approach such as NE is that if the plant-model mismatch cannot be fully explained by via a linearization of the model parameters, the converged point is not guaranteed to be the true optimum. This is evident from observing that the set points of the temperatures do not reach their optimal values, especially with the converged temperature of tray 14 being approximately 0.5°C from the true optimum. The reason for this discrepancy is due to the fact that this set point has limited influence on the objective and constraints of the system, something that has already been observed in other systems where a degree of freedom has a lower impact on the optimum (Papasavvas et al., 2017).

8. Conclusion

Modifier adaptation (MA) is an approach which guarantees that convergence occurs at the plant optimum provided accurate estimates of the plant gradients are obtained. This advantage has developed interest in the technique, with several variants being proposed which improve on the standard method. One such extension allows for the use of MA when optimizing a process with a difference in degrees of freedom between the plant and the model. This has be extended in this article to allow for the use of transient measurements in the real-time optimization strategy, speeding up convergence into a single smooth step.

The proposed scheme allows for the use of transient measurements via estimating the steady-state points using the current measurements of the plant. This led to issues with the plant gradient estimates as the standard method of using finite differences cannot be applied. This prompted the use of model-based gradient estimate techniques, such as neighboring extremals, to estimate the steady-state gradients of the plant without the need of many steady-state measurements. If the estimates of the gradients are accurate, the proposed method still converges to the plant optimum.

An issue was met upon using this technique of model-based gradient estimate for a system where there is a degree of freedom difference between the plant and model. This article has suggested a technique to allow for the use of an open-loop model to estimate the gradient of a controlled plant via some limited information about the relationship between the open-loop inputs to the plant i.e. where the dependencies are. This method allows for the rapid convergence to a near optimal point, with convergence times of the order of the settling time of the plant, which is significantly quicker than standard MA.

The proposed scheme was illustrated on a case study of a CSTR, in which the scheme quickly converged to a near optimal solution in the order of a single steady-state convergence time. This was followed by the more challenging case study of a distillation column for the separation of methanol from n-propanol. The scheme also rapidly converged to a near optimal point in the order of a couple of settling times, and could track a change in the optimal conditions well, despite the highly non-linear nature of the problem.

Funding

This work was supported by the Engineering and Physical Sciences Research Council [grant number R513209].

References

Ben-Israel, A., Greville, T.N.E., Society, C.M., 2003. Generalized Inverses: Theory and Applications. CMS Books in Mathematics, Springer.

Brdyś, M., Tatjewski, P., 1994. An Algorithm for Steady-State Optimizing Dual Control of Uncertain Plants. IFAC Proceedings Volumes 27, 215–220.

Cadavid, J., Hernández, R., Engell, S., 2017. Speed-up of Iterative Real-Time Optimization by Estimating the Steady States in the Transient Phase using Nonlinear System Identification. IFAC-PapersOnLine .

Câmara, M.M., Quelhas, A.D., Pinto, J.C., 2016. Performance evaluation of real industrial RTO systems. Processes 4, 44.

Costello, S., François, G., Bonvin, D., 2013. Real-time optimization when plant and model have different sets of inputs, in: IFAC Proceedings Volumes (IFAC-PapersOnline), pp. 39–44.

Costello, S., François, G., Bonvin, D., 2016. A Directional Modifier-Adaptation Algorithm for Real-Time Optimization. Journal of Process Control 39, 64–76.

Darby, M.L., Nikolaou, M., Jones, J., Nicholson, D., 2011. RTO: An overview and assessment of current practice.

Diehl, M., Uslu, I., Findeisen, R., Schwarzkopf, S., Allgöwer, F., Bock, H.G., Bürner, T., Gilles, E.D., Kienle, A., Schlöder, J.P., Stein, E., 2001. Real-Time Optimization for Large Scale Processes: Nonlinear Model Predictive Control of a High Purity Distillation Column, in: Online Optimization of Large Scale Systems. Springer Berlin Heidelberg, pp. 363–383.

Dunford, N., 1988. Linear Operators. Wiley classics library, John Wiley.

Ferreira, T.d.A., François, G., Marchetti, A.G., Bonvin, D., 2017. Use of Transient Measurements for Static Real-Time Optimization. IFAC-PapersOnLine 50, 5737–5742.

François, G., Bonvin, D., 2013. Use of convex model approximations for real-time optimization via modifier adaptation. Industrial and Engineering Chemistry Research 52, 11614–11625.

François, G., Bonvin, D., 2014. Use of transient measurements for the optimization of steady-state performance via modifier adaptation. Industrial and Engineering Chemistry Research 53, 5148–5159.

François, G., Costello, S., Marchetti, A.G., Bonvin, D., 2016. Extension of modifier adaptation for controlled plants using static open-loop models. Computers and Chemical Engineering 93, 361–371.

François, G., Srinivasan, B., Bonvin, D., 2012. Comparison of six implicit real-time optimization schemes. Journal Europeen des Systemes Automatises 46, 291–305.

Gao, W., Engell, S., 2005. Iterative set-point optimization of batch chromatography. Computers & Chemical Engineering 29, 1401–1409.

Gao, W., Engell, S., 2016. Using Transient Measurements in Iterative Steady-State Optimizing Control, in: Computer Aided Chemical Engineering. volume 38, pp. 511–516.

Gao, W., Wenzel, S., Engell, S., 2016. A reliable modifier-adaptation strategy for real-time optimization. Computers and Chemical Engineering 91, 318–328.

Gros, S., Srinivasan, B., Bonvin, D., 2009. Optimizing control based on output feedback. Computers and Chemical Engineering 33, 191–198.

Jang, S.S., Joseph, B., Mukai, H., 1987. On-line optimization of constrained multivariable chemical processes. AIChE Journal 33, 26–35.

Marchetti, A., Chachuat, B., Bonvin, D., 2009. Modifier-adaptation methodology for real-time optimization. Industrial and Engineering Chemistry Research 48, 6022–6033.

Marchetti, A., Chachuat, B., Bonvin, D., 2010. A dual modifier-adaptation approach for real-time optimization, in: Journal of Process Control, Elsevier. pp. 1027–1037.

Navia, D., Briceño, L., Gutiérrez, G., De Prada, C., 2015. Modifier-Adaptation Methodology for Real-Time Optimization Reformulated as a Nested Optimization Problem. Industrial and Engineering Chemistry Research 54, 12054–12071.

Papasavvas, A., de Avila Ferreira, T., Marchetti, A.G., Bonvin, D., 2017. Real-Time Optimization via Modifier Adaptation using Partial Plant Models. IFAC-PapersOnLine 50, 4666–4671.

Quelhas, A.D., de Jesus, N.J.C., Pinto, J.C., 2013. Common vulnerabilities of RTO implementations in real chemical processes. Canadian Journal of Chemical Engineering 91, 652–668.

Rao, C.R., Mitra, S.K., 1971. Generalized Inverse of Matrices and Its Applications. Probability and Statistics Series, Wiley.

Roberts, P.D., 1979. An algorithm for steady-state system optimization and parameter estimation. International Journal of Systems Science 10, 719–734.

Rodríguez-Blanco, T., Sarabia, D., Pitarch, J., de Prada, C., 2017. Modifier Adaptation methodology based on transient and static measurements for RTO to cope with structural uncertainty. Computers & Chemical Engineering 106, 480–500.

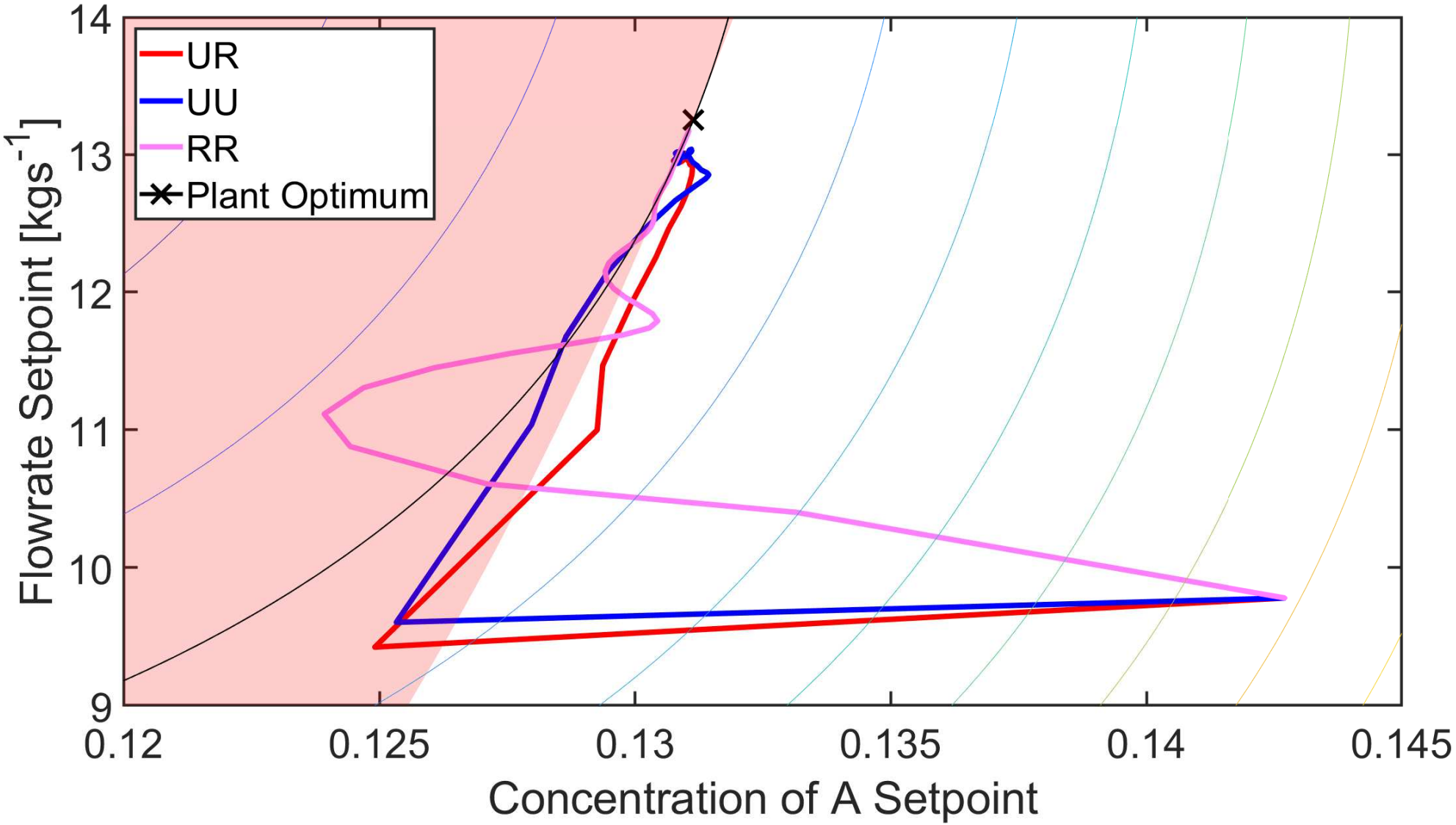
Srinivasan, B., 2007. Real-time optimization of dynamic systems using multiple units. International Journal of Robust and Nonlinear Control 17, 1183–1193.

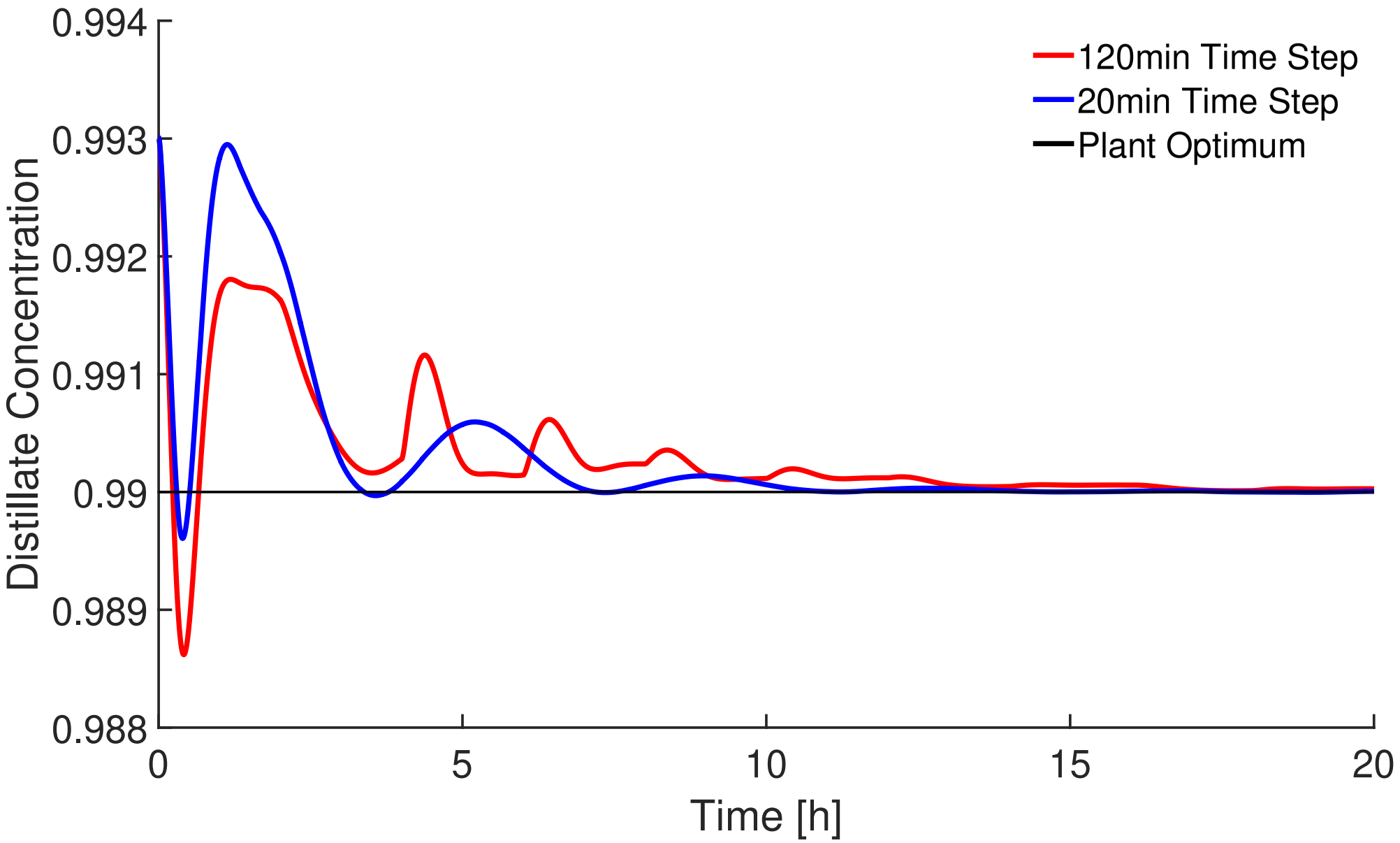
Tatjewski, P., 2002. Iterative optimizing set-point control - The basic principle redesigned, in: IFAC Proceedings Volumes (IFAC-PapersOnline), Elsevier. pp. 49–54.

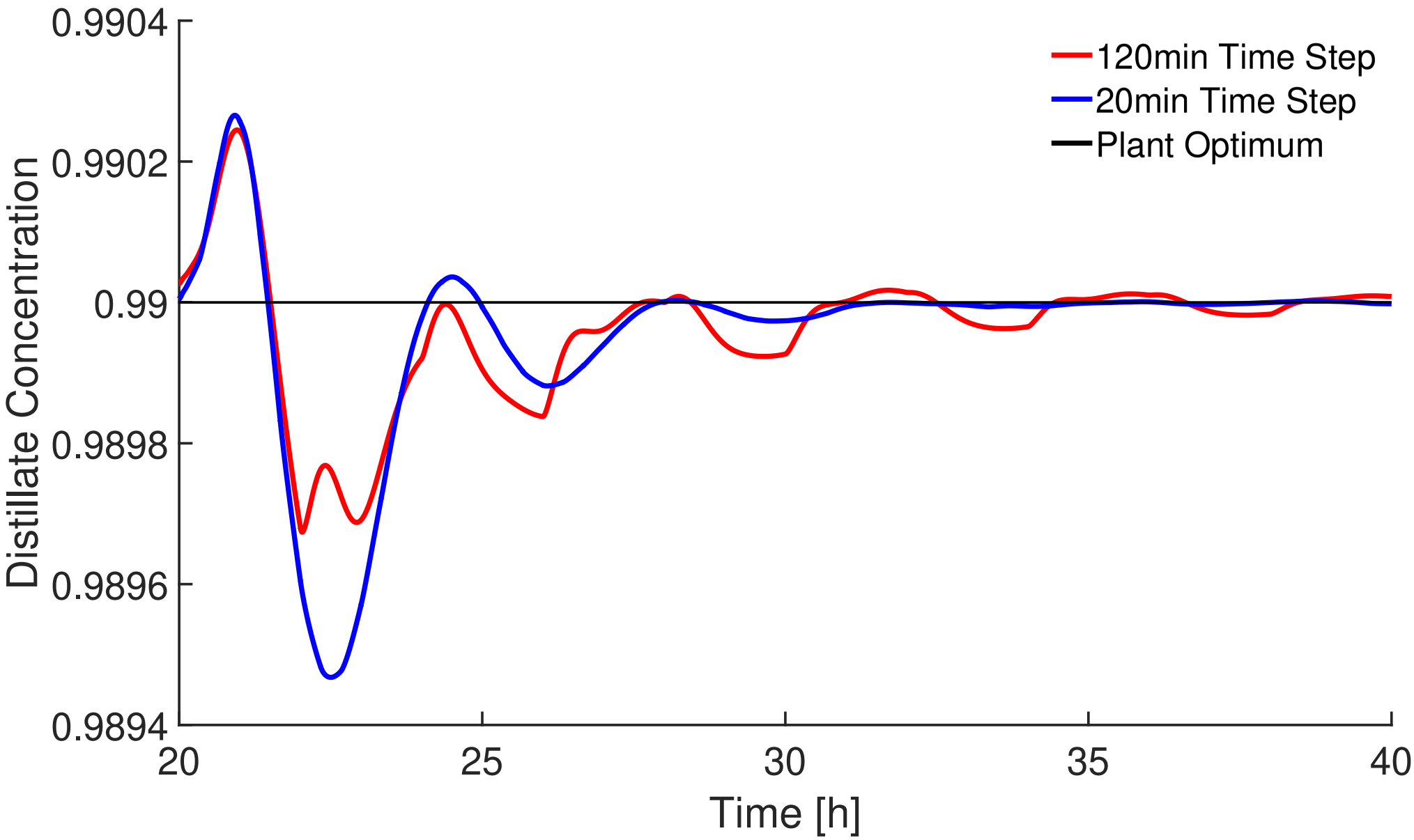
Vahidi, A., Stefanopoulou, A., Peng, H., 2005. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: Theory and experiments. Vehicle System Dynamics 43, 31–55.

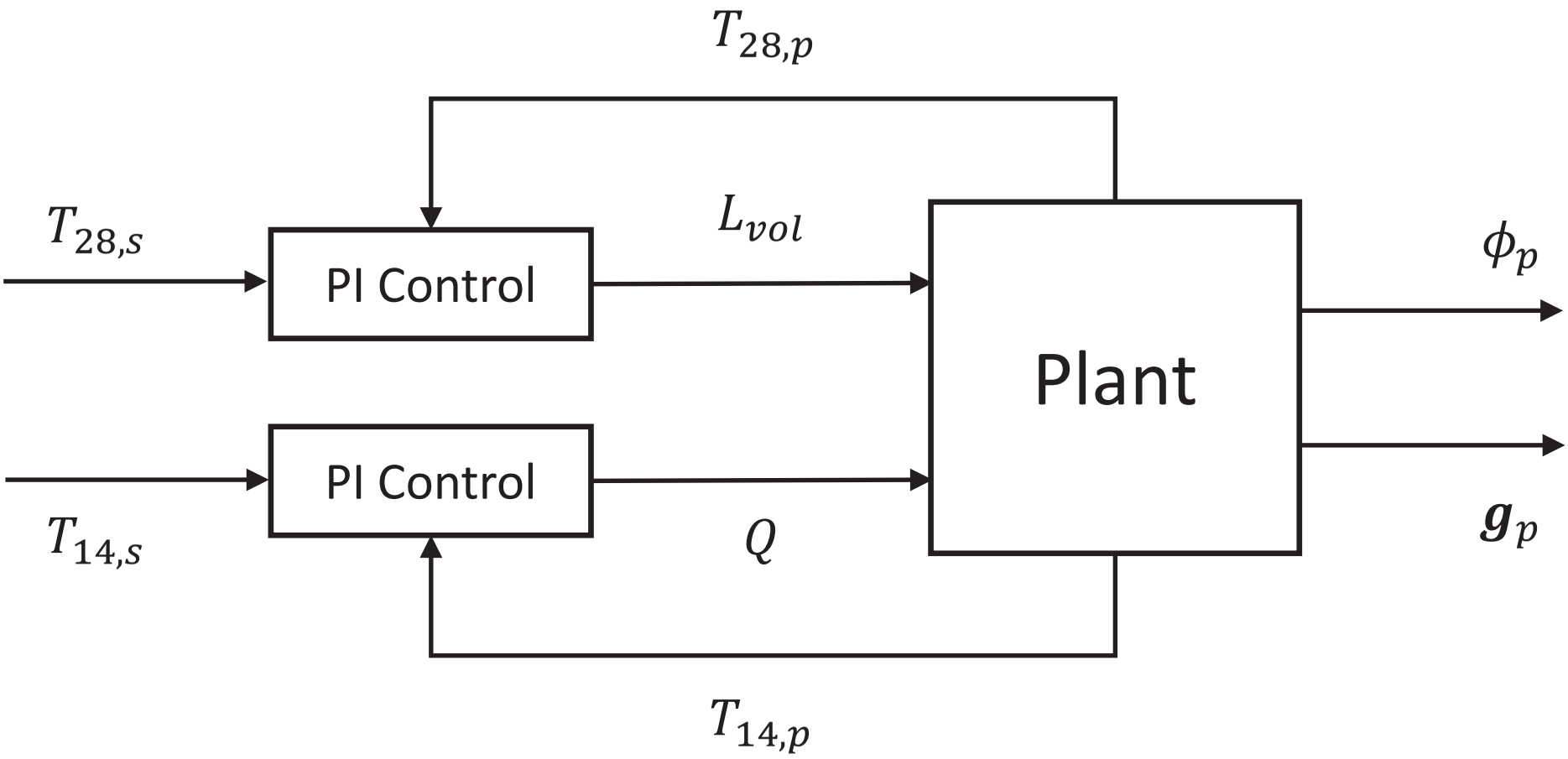
Williams, T.J., Otto, R.E., 1960. A generalized chemical processing model for the investigation of computer control. Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics 79, 458–473.

Yip, W.S., Marlin, T.E., 2004. The effect of model fidelity on real-time optimization performance, in: Computers and Chemical Engineering, Elsevier Ltd. pp. 267–280.

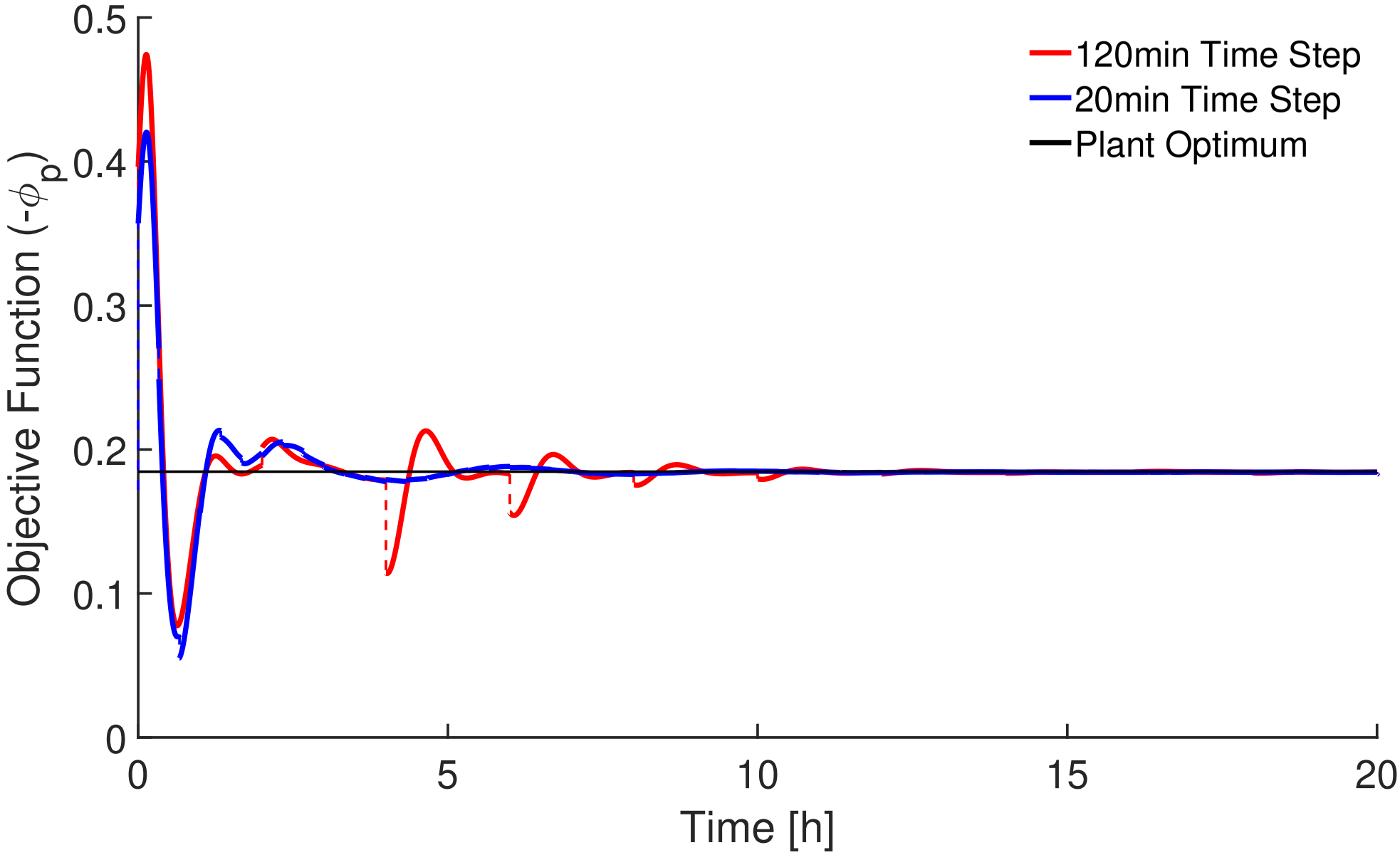

UR
UU
RR
Plant Optimum
Flowrate Setpoint [kgs-1]
Concentration of A Setpoint
14
13
12
11
10
9
0.12
0.125
0.13
0.135
0.14
0.145

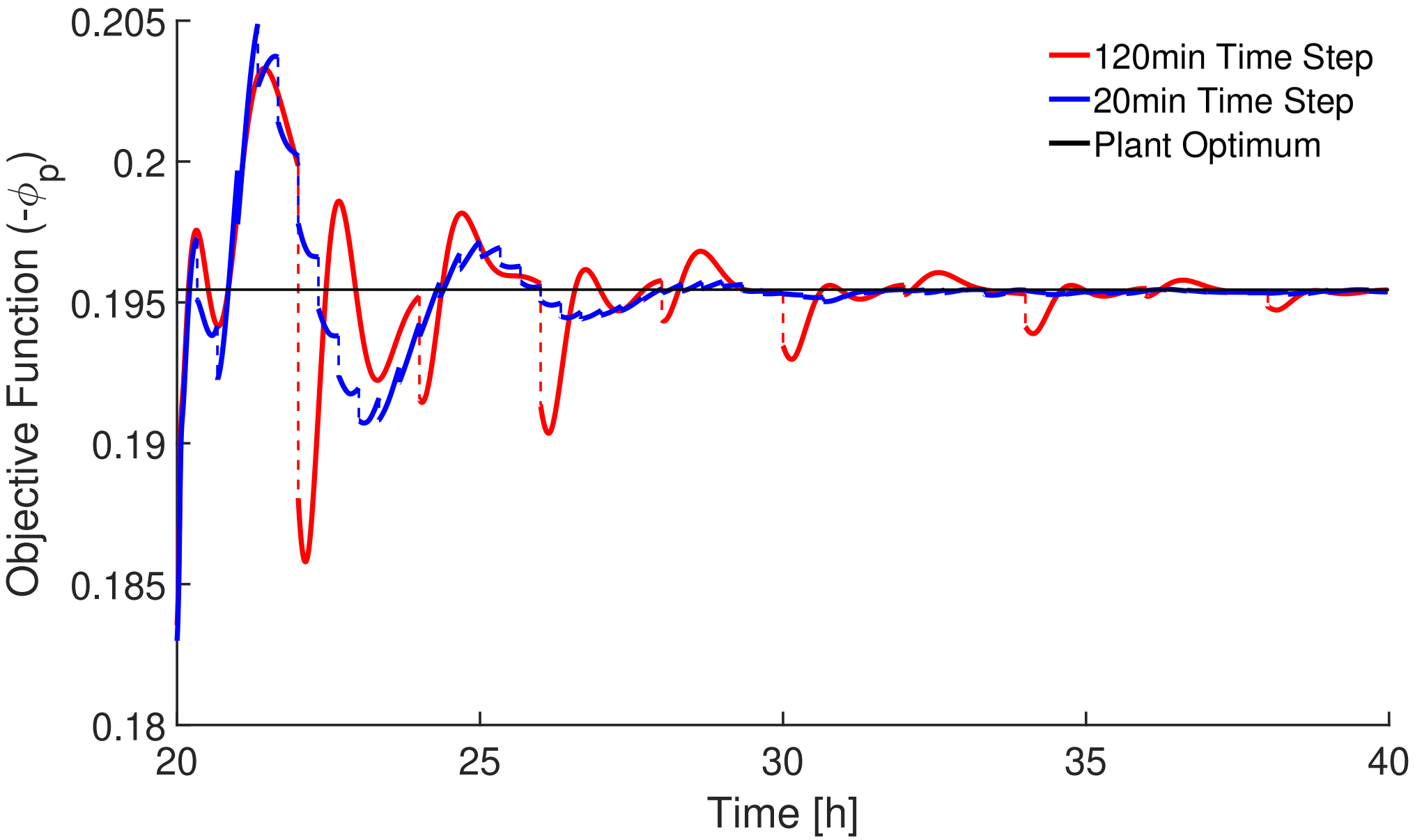

0.994
0.993
0.992
0.991
0.99
0.989
0.988
Distillate Concentration
0
5
10
15
20
Time [h]
120min Time Step
20min Time Step
Plant Optimum

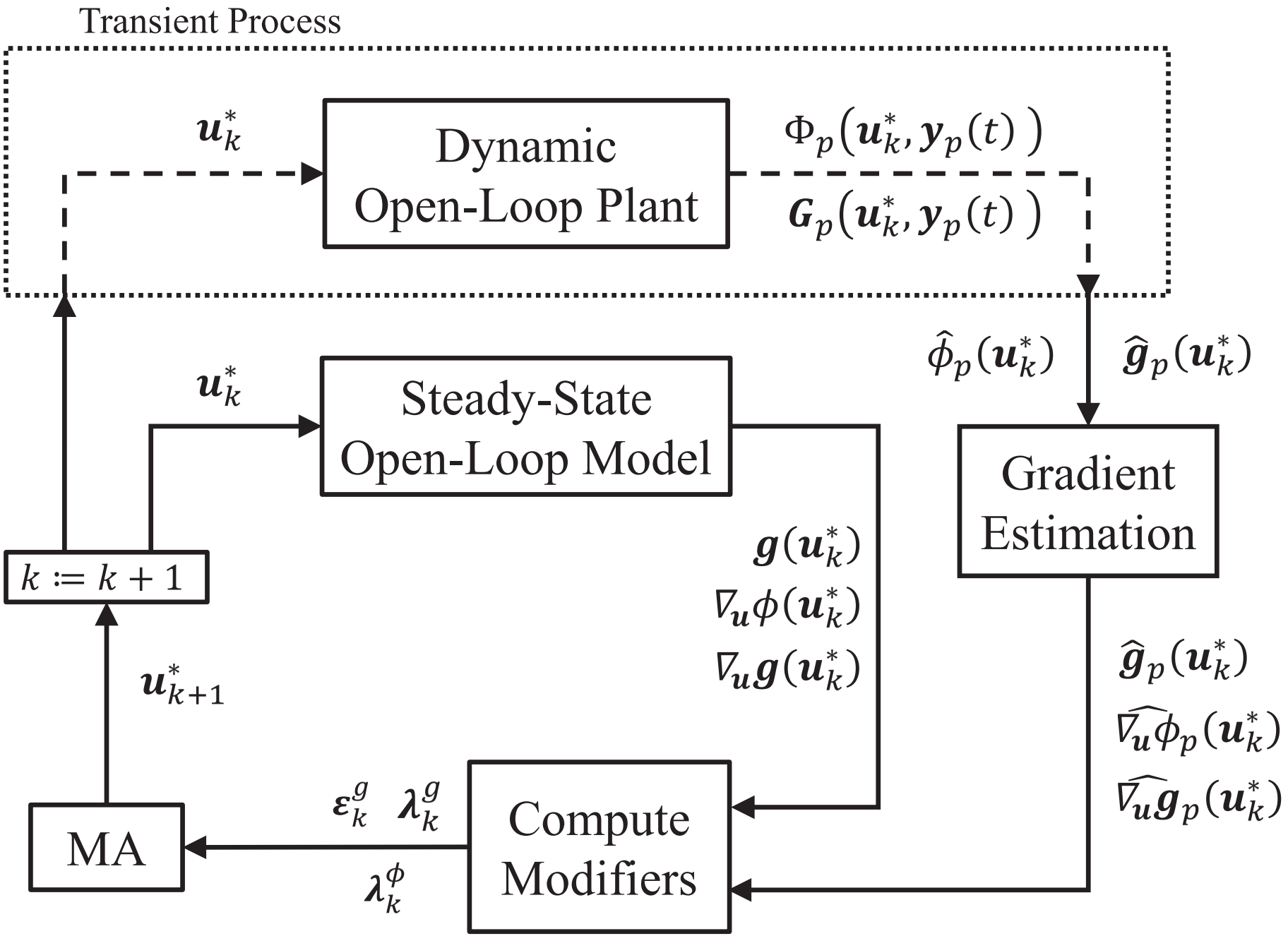

0.9904
0.9902
0.99
0.9898
0.9896
0.9894
Distillate Concentration
20
25
30
35
40
Time [h]
120min Time Step
20min Time Step
Plant Optimum

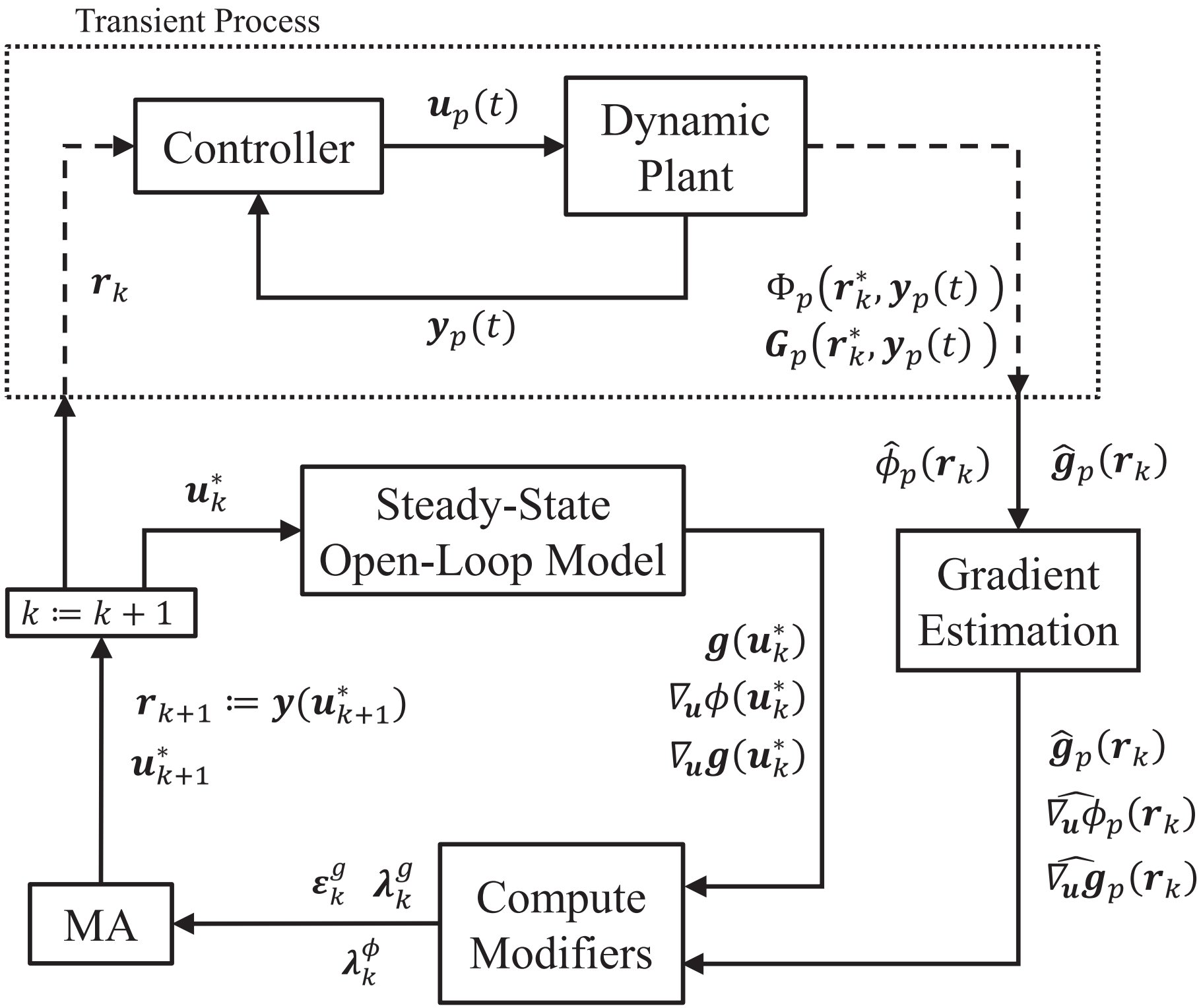

$T_{28,p}$
$T_{28,s}$
PI Control
L_{vol}
ϕ_p
Plant
PI Control
$T_{14,s}$
Q
g_p
$T_{14,p}$

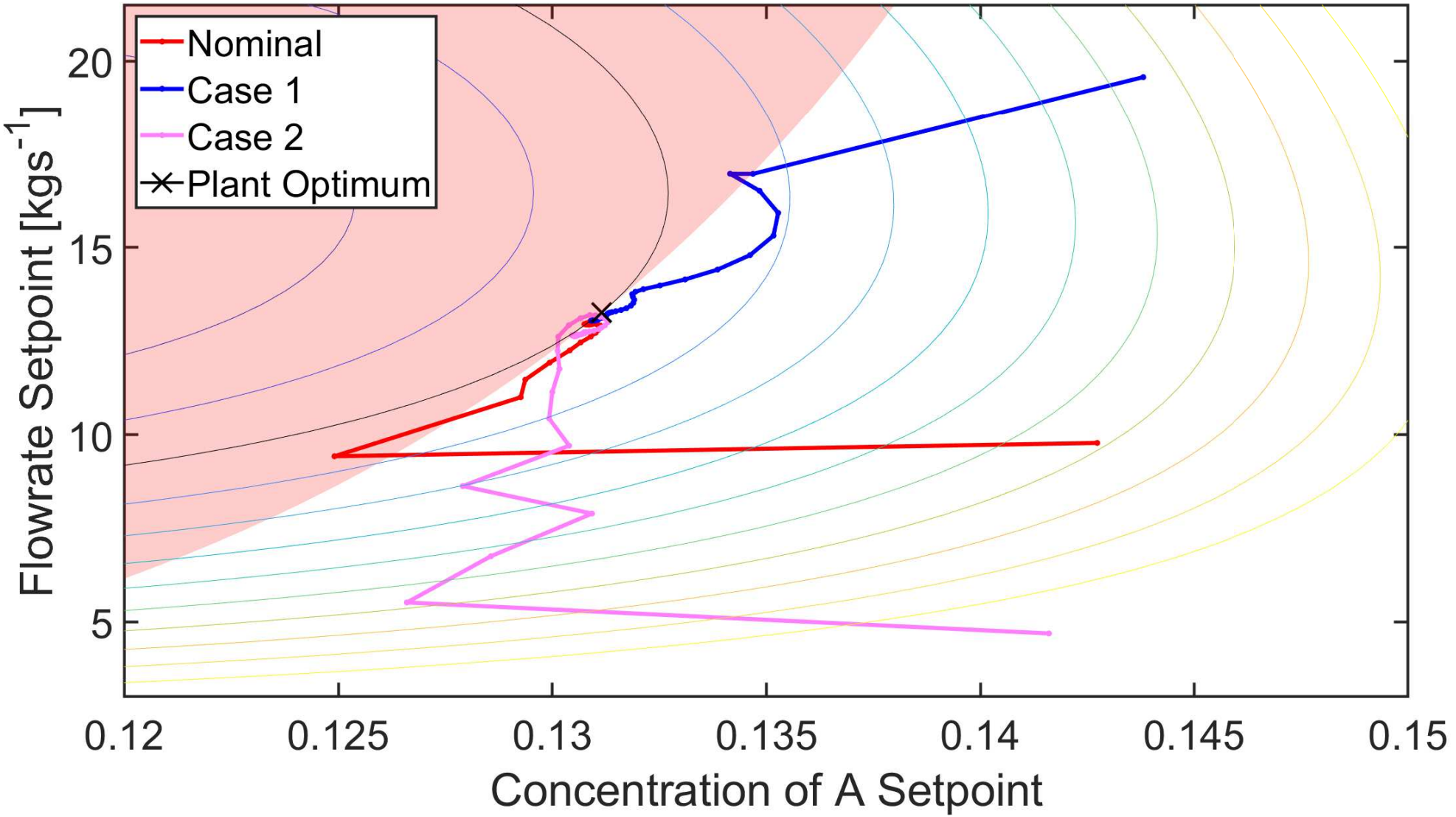
Model
n
$g(n)$
$\phi(n)$
Controller
Plant
r
d_n
d_y
$G_p(r)$
$\phi_p(r)$

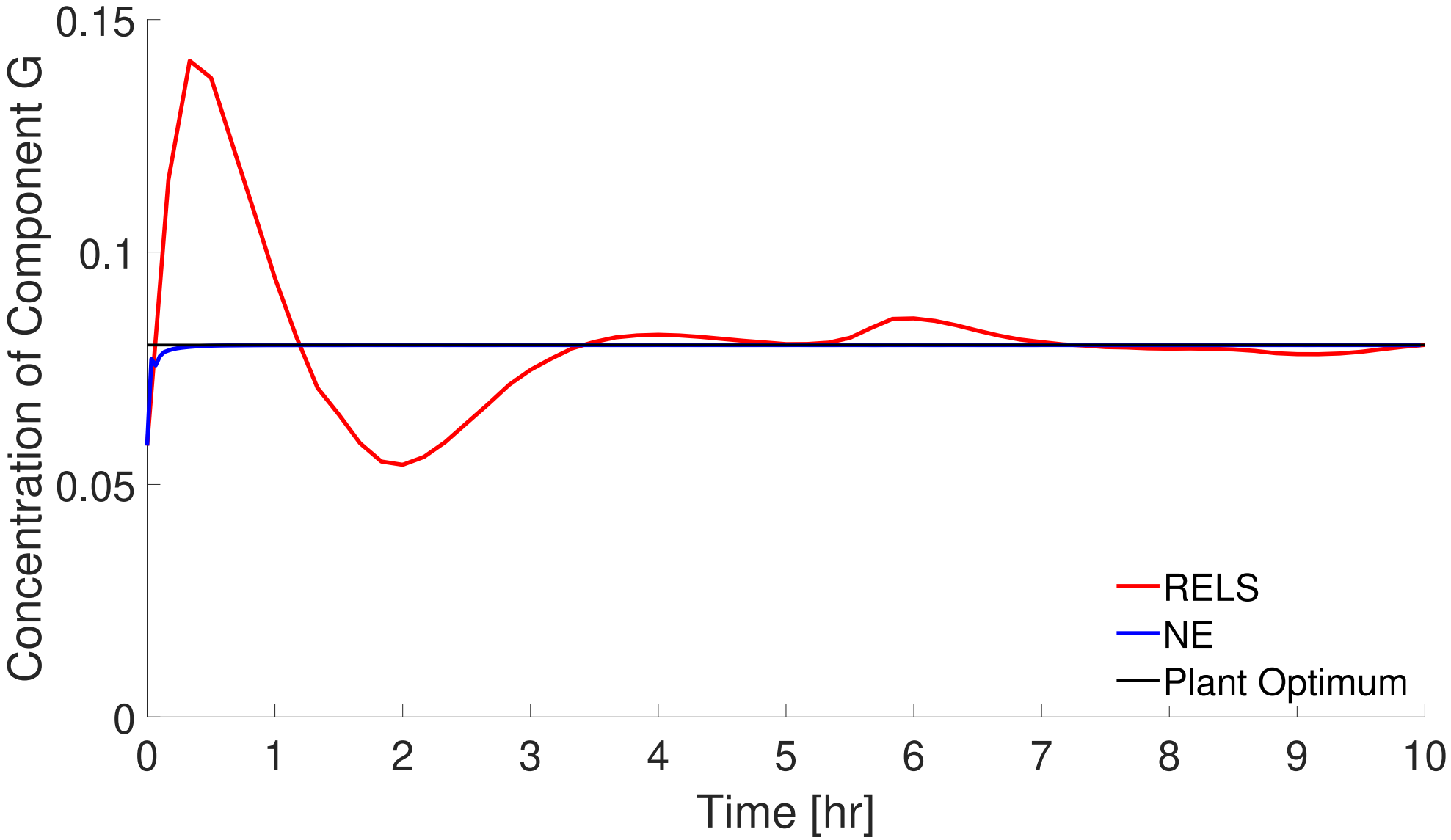

0.5
0.4
0.3
0.2
0.1
0
Objective Function (-ϕ_p)
0
5
10
15
20
Time [h]
120min Time Step
20min Time Step
Plant Optimum

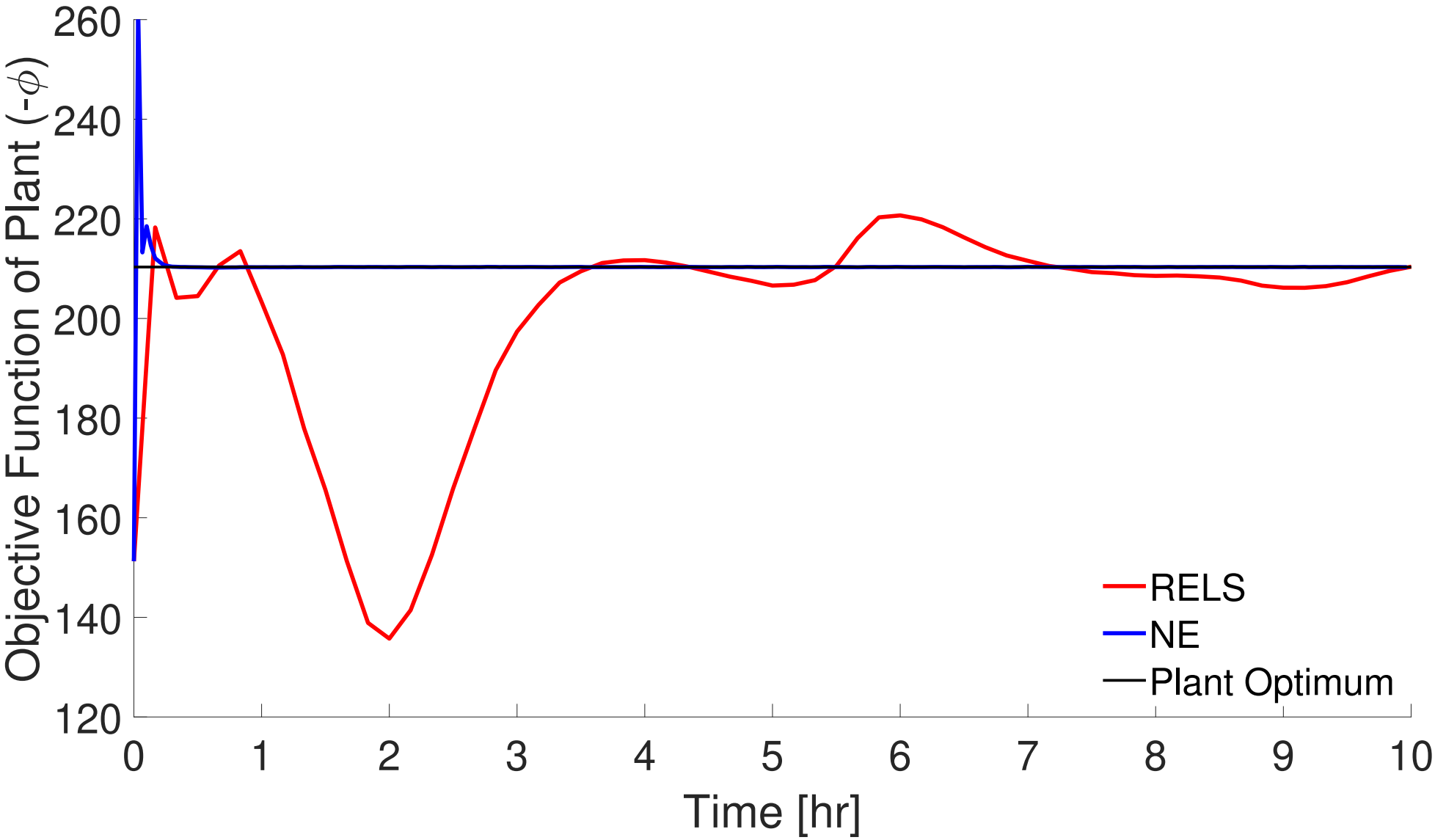

0.205
0.2
0.195
0.19
0.185
0.18
Objective Function (-ϕ_p)
20
25
30
35
40
Time [h]
120min Time Step
20min Time Step
Plant Optimum

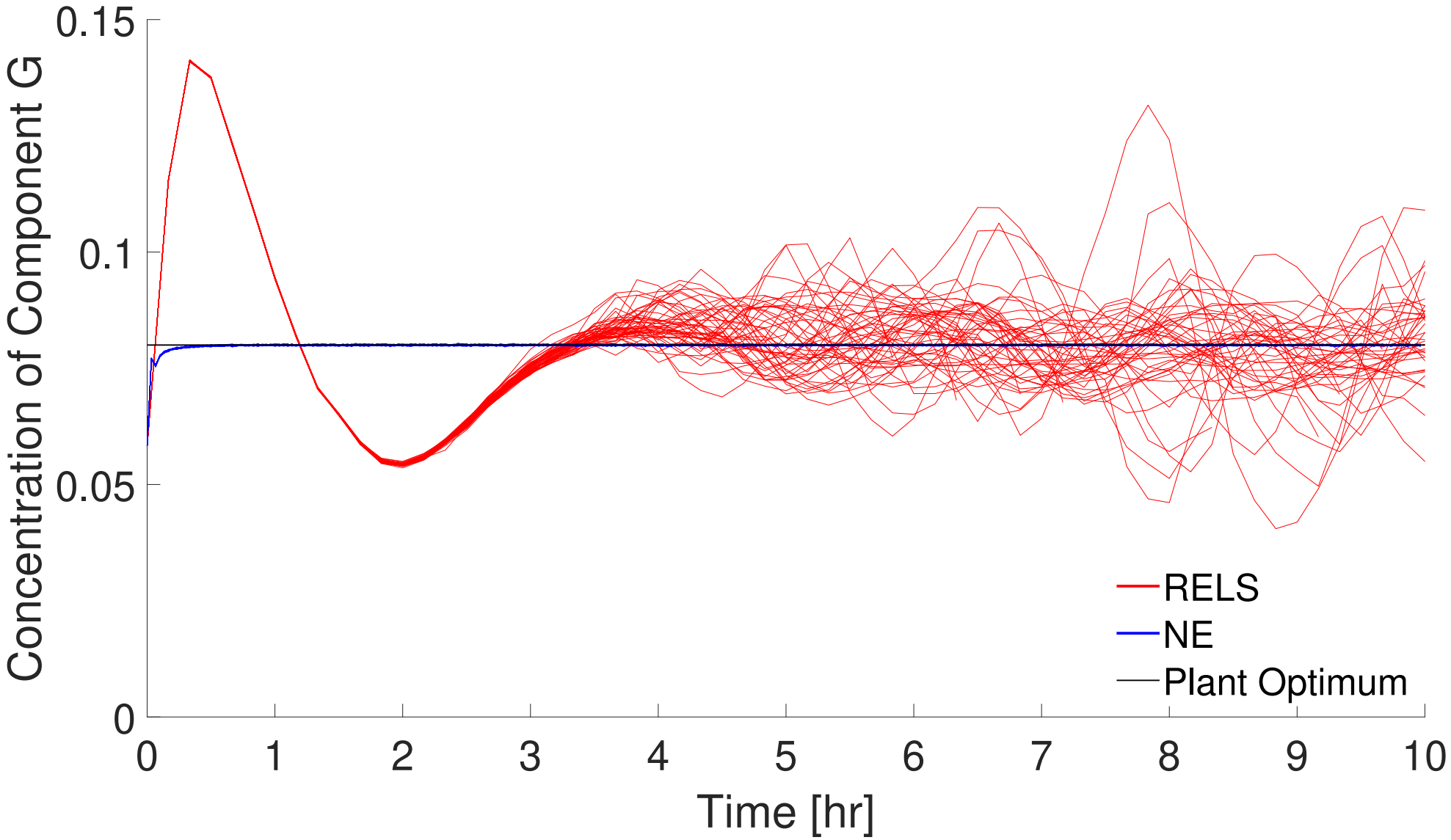

Transient Process
$\boldsymbol{u}_k^*$
Dynamic
Open-Loop Plant
$\Phi_p(\boldsymbol{u}_k^*, \boldsymbol{y}_p(t))$
$\boldsymbol{G}_p(\boldsymbol{u}_k^*, \boldsymbol{y}_p(t))$
$\hat{\phi}_p(\boldsymbol{u}_k^*)$
$\hat{\boldsymbol{g}}_p(\boldsymbol{u}_k^*)$
$\boldsymbol{u}_k^*$
Steady-State
Open-Loop Model
Gradient
Estimation
$k := k + 1$
$\boldsymbol{g}(\boldsymbol{u}_k^*)$
$\nabla_{\boldsymbol{u}}\phi(\boldsymbol{u}_k^*)$
$\nabla_{\boldsymbol{u}}\boldsymbol{g}(\boldsymbol{u}_k^*)$
$\hat{\boldsymbol{g}}_p(\boldsymbol{u}_k^*)$
$\widehat{\nabla_{\boldsymbol{u}}\phi_p}(\boldsymbol{u}_k^*)$
$\widehat{\nabla_{\boldsymbol{u}}\boldsymbol{g}_p}(\boldsymbol{u}_k^*)$
$\boldsymbol{u}_{k+1}^*$
$\boldsymbol{\varepsilon}_k^g$ $\boldsymbol{\lambda}_k^g$
$\boldsymbol{\lambda}_k^\phi$
MA
Compute
Modifiers

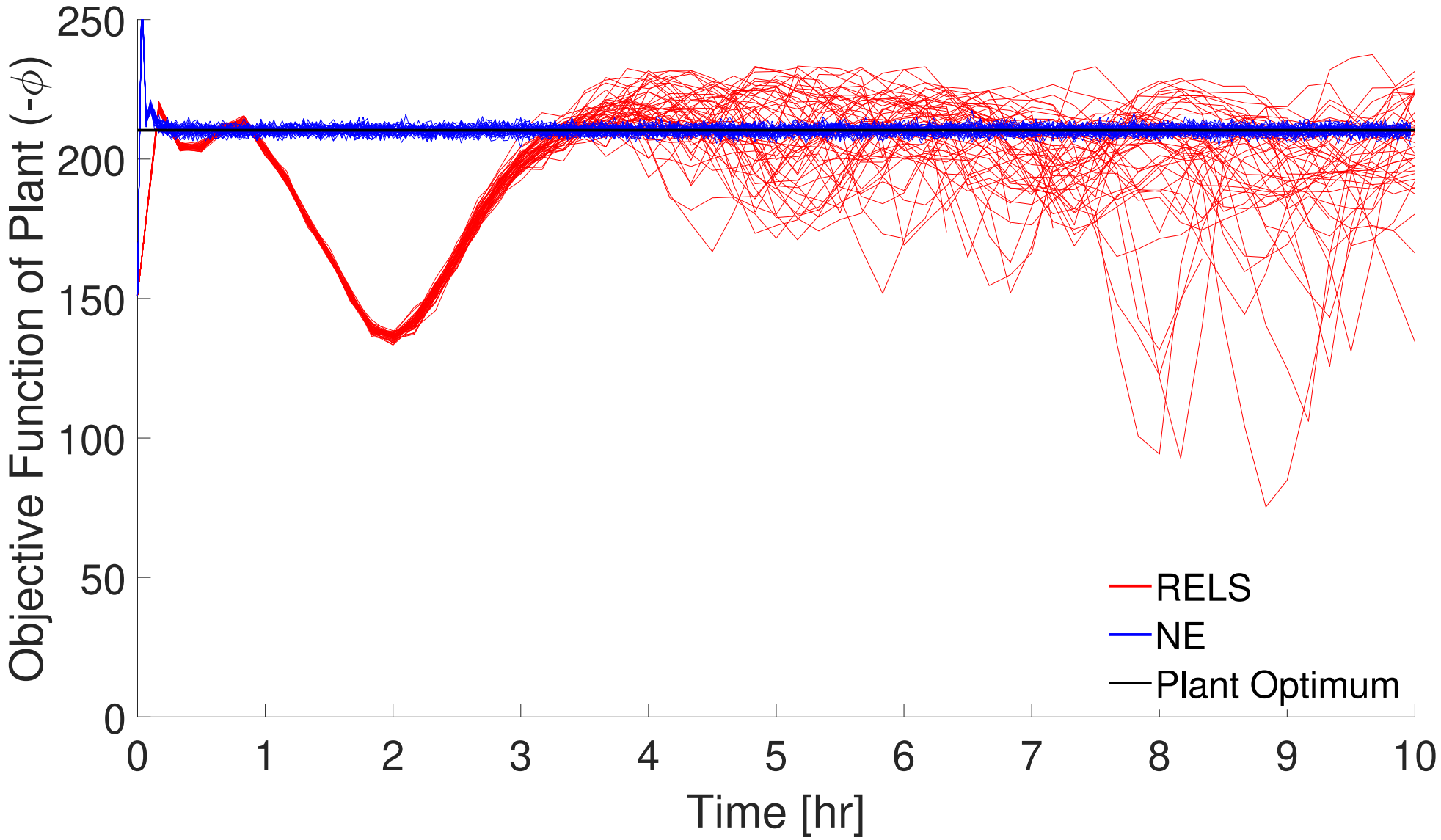

Transient Process
Controller
$u_p(t)$
Dynamic Plant
r_k
$y_p(t)$
$\Phi_p(r_k^*, y_p(t))$
$G_p(r_k^*, y_p(t))$
$\hat{\phi}_p(r_k)$
$\hat{g}_p(r_k)$
u_k^*
Steady-State Open-Loop Model
Gradient Estimation
$k := k+1$
$g(u_k^*)$
$\nabla_u \phi(u_k^*)$
$\nabla_u g(u_k^*)$
$r_{k+1} := y(u_{k+1}^*)$
u_{k+1}^*
$\hat{g}_p(r_k)$
$\widehat{\nabla_u \phi_p}(r_k)$
$\widehat{\nabla_u g_p}(r_k)$
ε_k^g λ_k^g
MA
Compute Modifiers
λ_k^ϕ

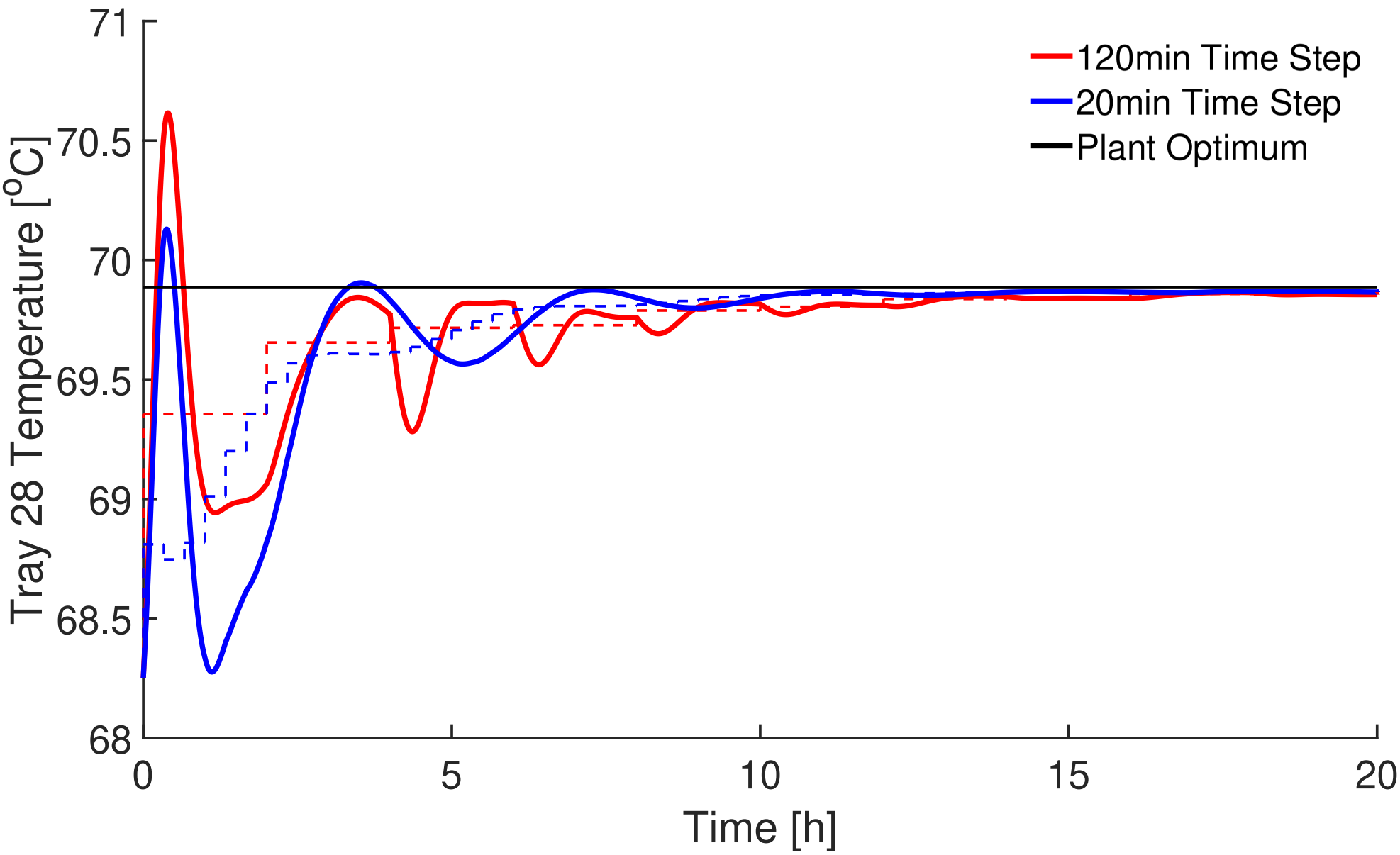

Nominal
Case 1
Case 2
Plant Optimum
Flowrate Setpoint [kgs-1]
20
15
10
5
0.12
0.125
0.13
0.135
0.14
0.145
0.15
Concentration of A Setpoint

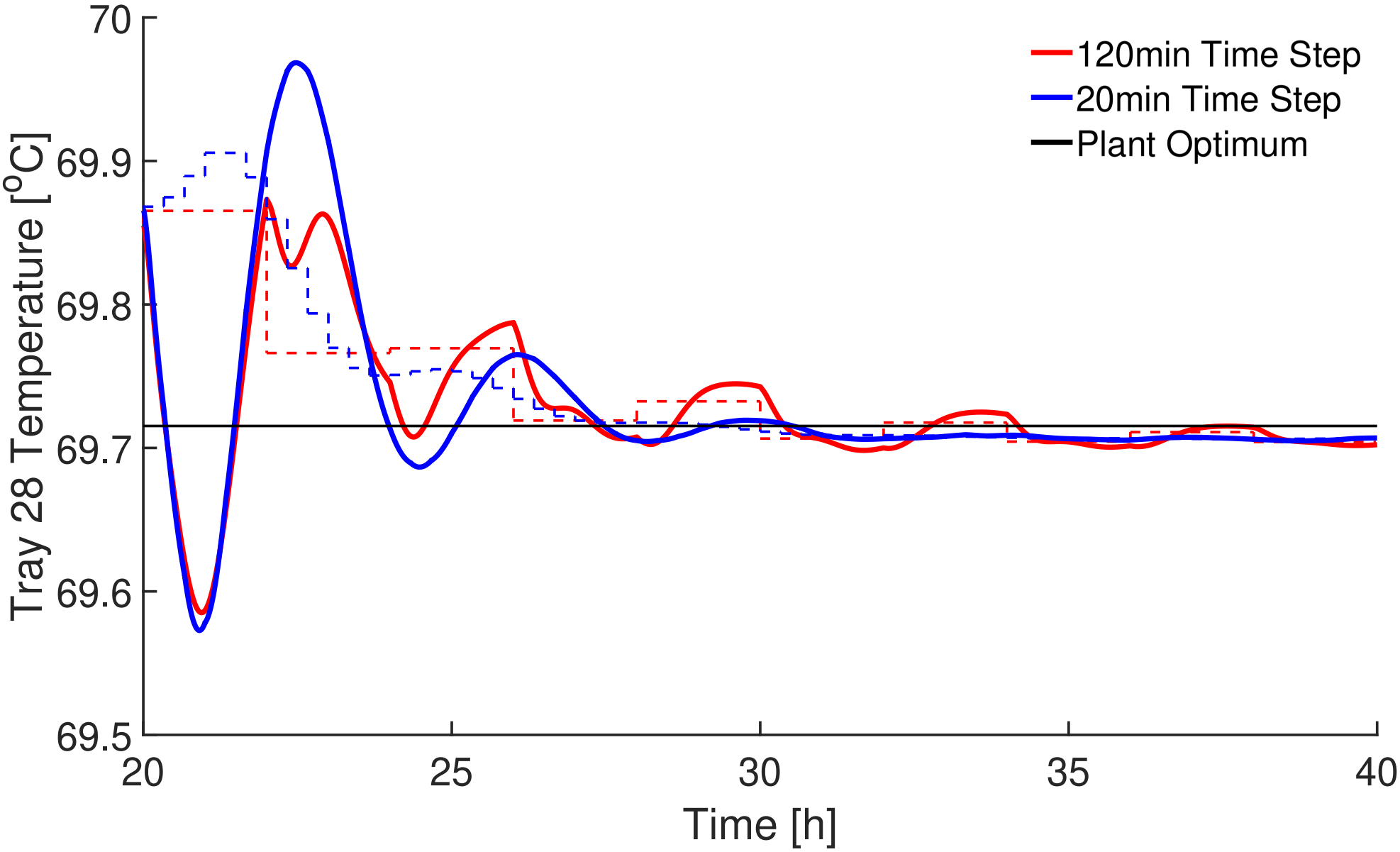

0.15
0.1
0.05
0
Concentration of Component G
0
1
2
3
4
5
6
7
8
9
10
Time [hr]
RELS
NE
Plant Optimum

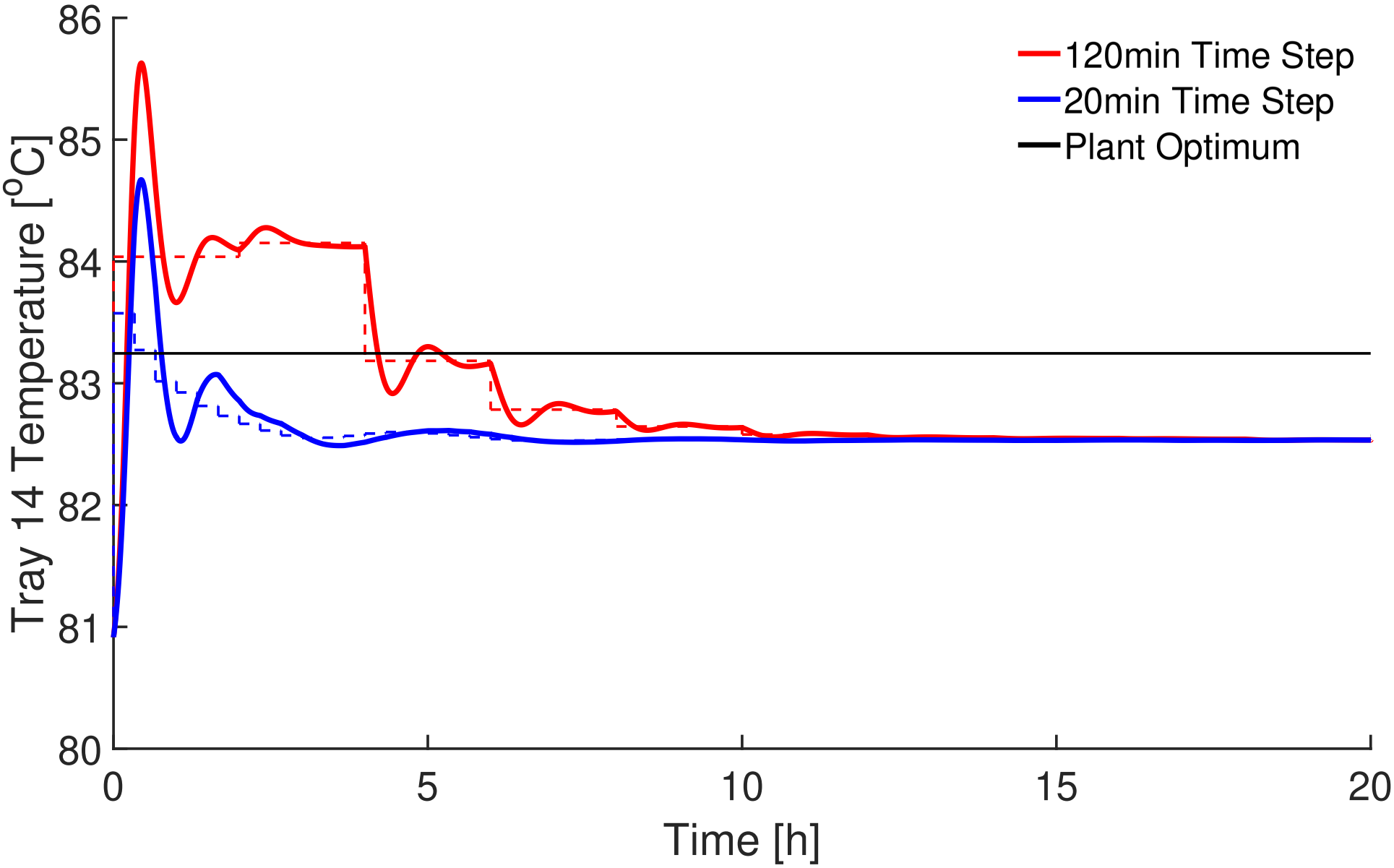

Objective Function of Plant (-ϕ)
260
240
220
200
180
160
140
120
0
1
2
3
4
5
6
7
8
9
10
Time [hr]
RELS
NE
Plant Optimum

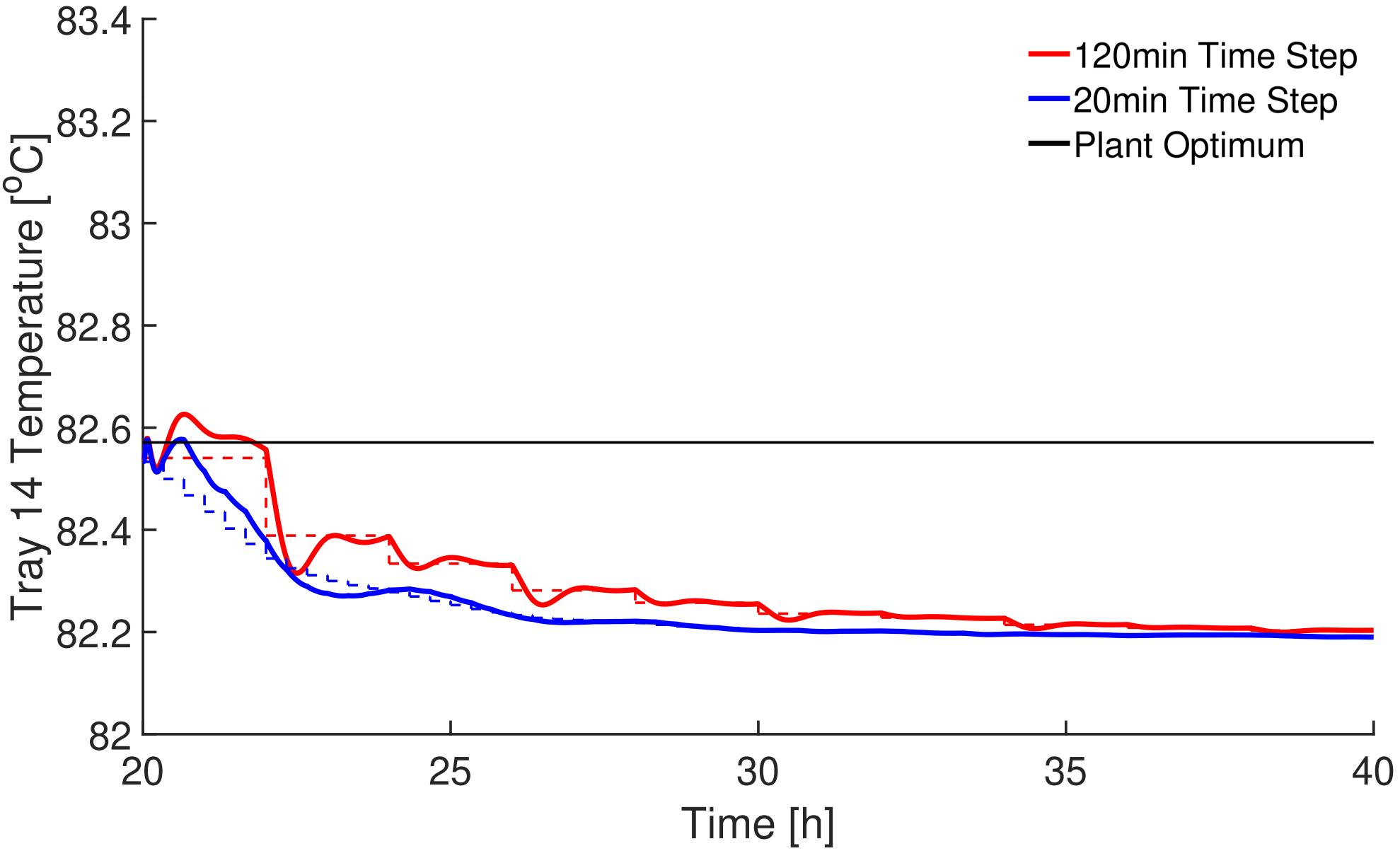

0.15
0.1
0.05
0
Concentration of Component G
0
1
2
3
4
5
6
7
8
9
10
Time [hr]
RELS
NE
Plant Optimum

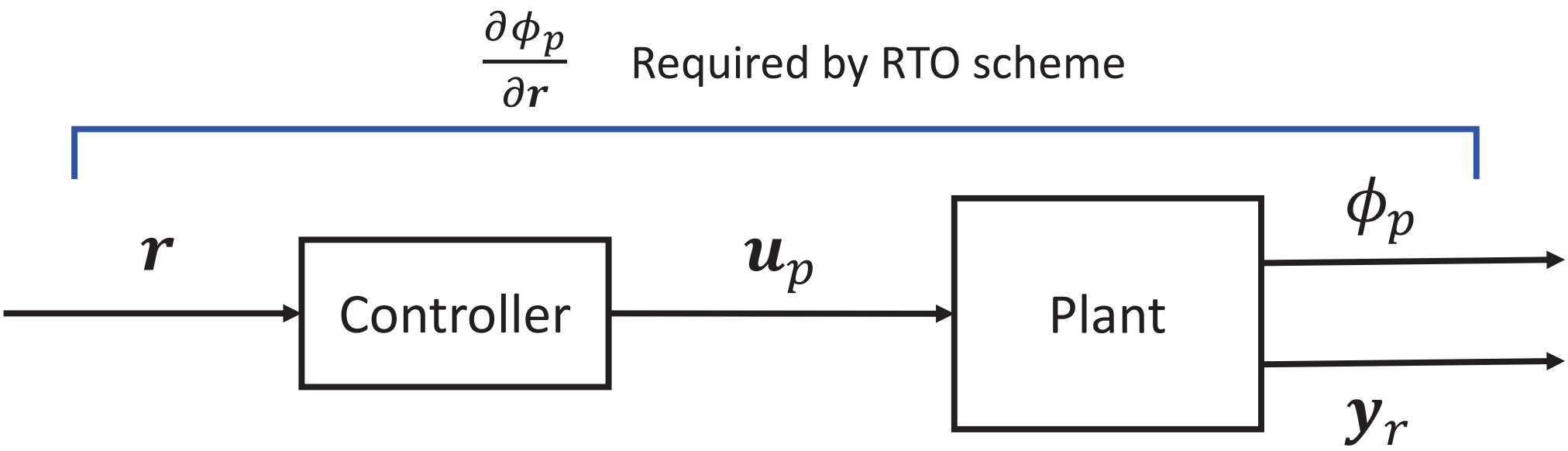

Objective Function of Plant (-ϕ)
250
200
150
100
50
0
0
1
2
3
4
5
6
7
8
9
10
Time [hr]
RELS
NE
Plant Optimum

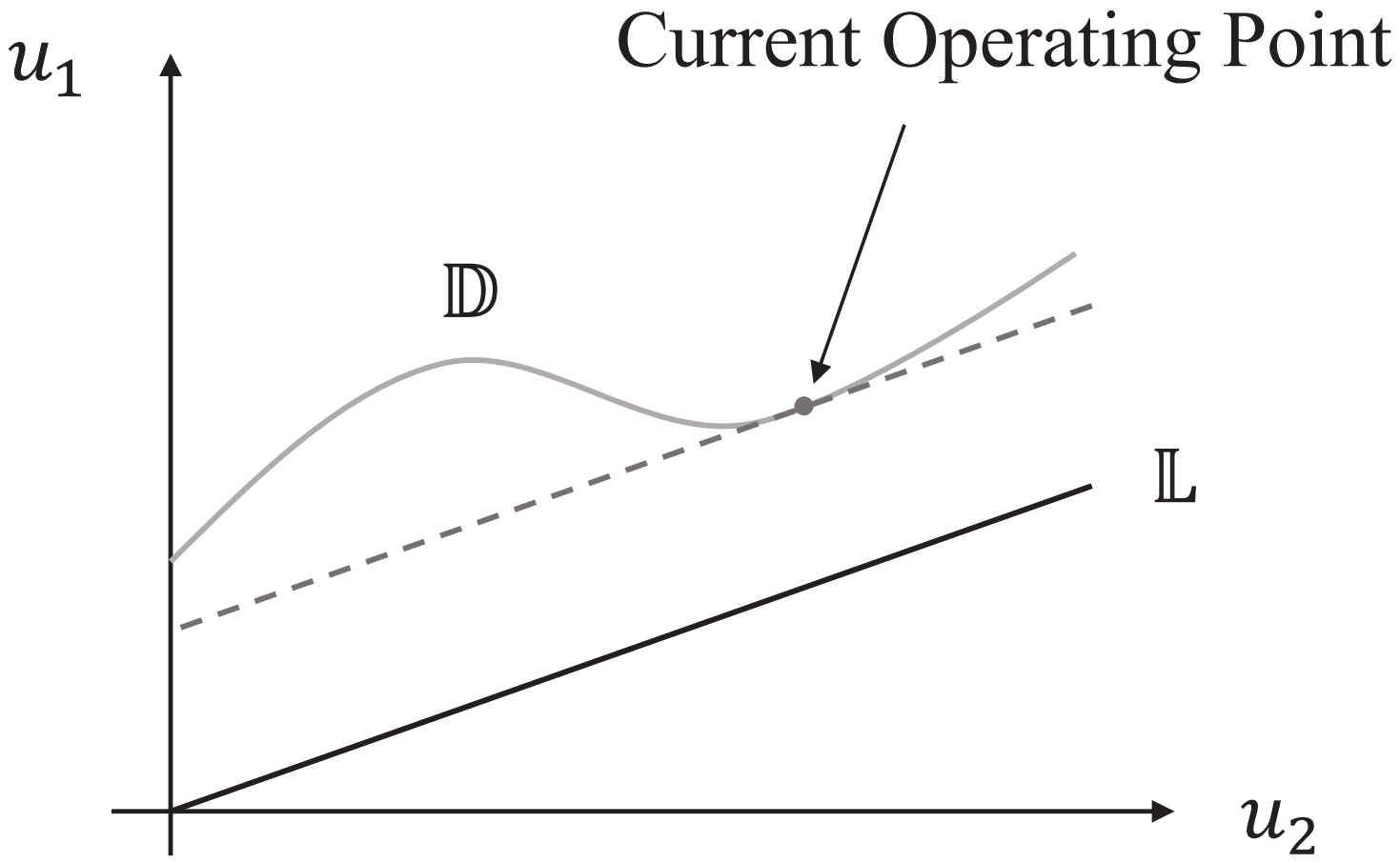

120min Time Step
20min Time Step
Plant Optimum
Tray 28 Temperature [°C]
Time [h]
71
70.5
70
69.5
69
68.5
68
0
5
10
15
20

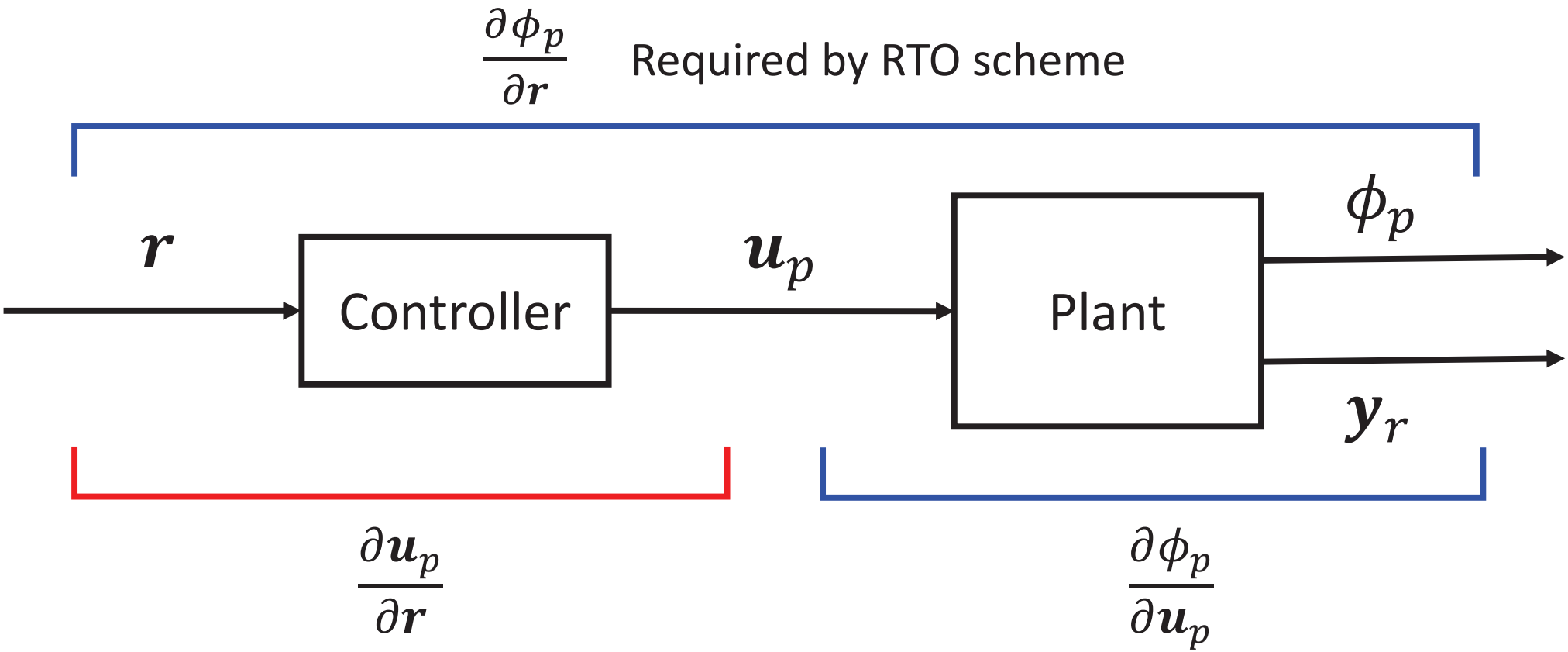

70
69.9
69.8
69.7
69.6
69.5
Tray 28 Temperature [°C]
20
25
30
35
40
Time [h]
120min Time Step
20min Time Step
Plant Optimum

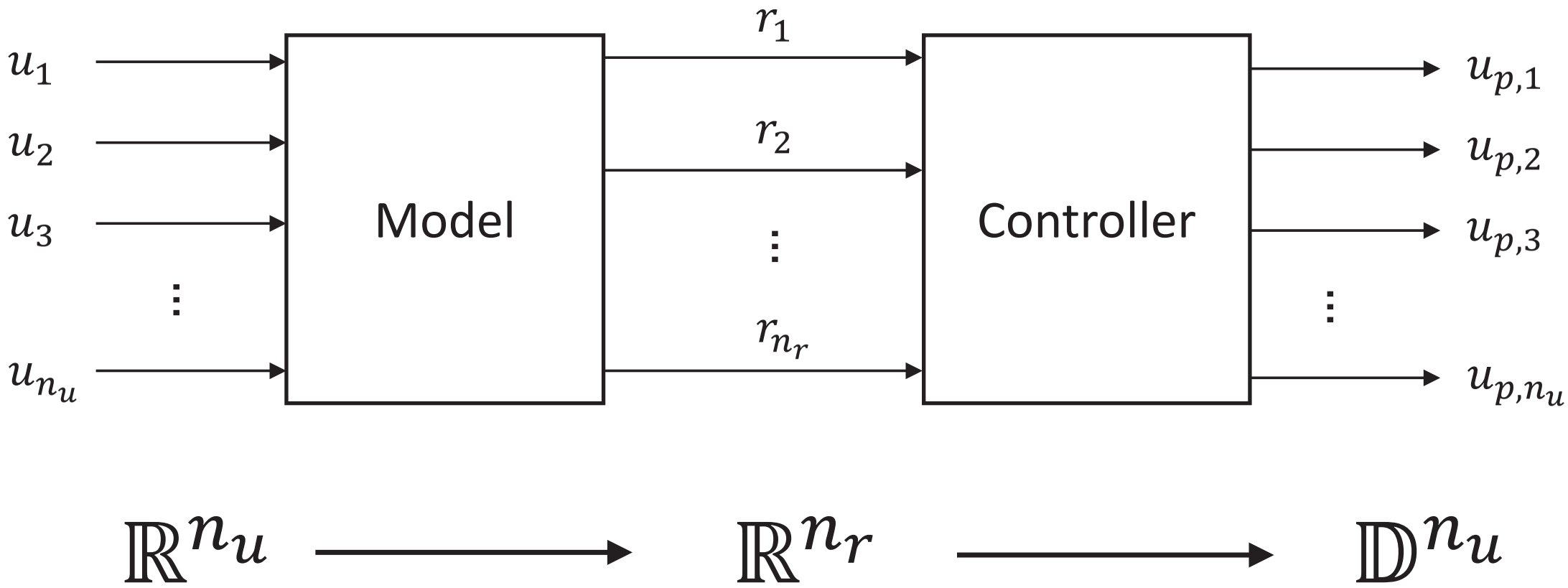

120min Time Step
20min Time Step
Plant Optimum
Tray 14 Temperature [°C]
Time [h]
86
85
84
83
82
81
80
0
5
10
15
20

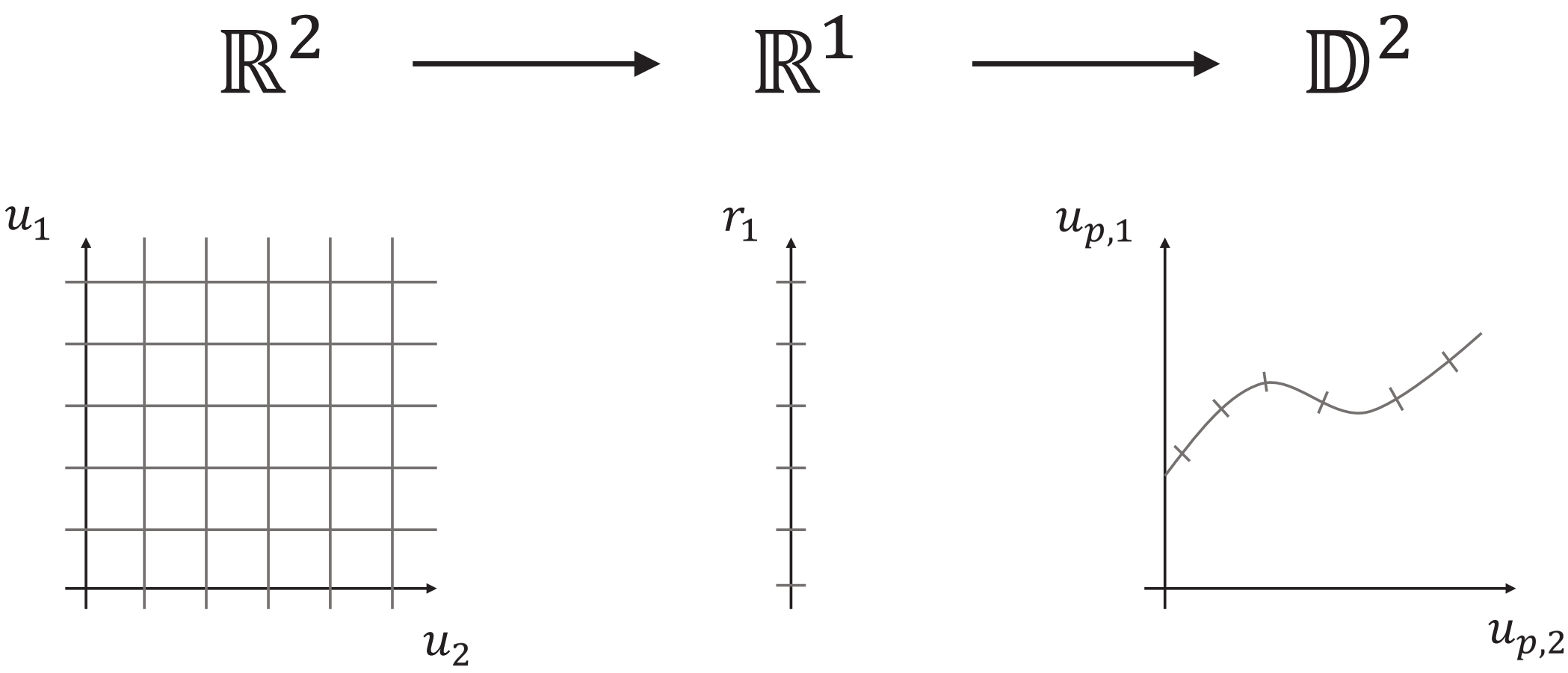

120min Time Step
20min Time Step
Plant Optimum
Tray 14 Temperature [°C]
83.4
83.2
83
82.8
82.6
82.4
82.2
82
20
25
30
35
40
Time [h]

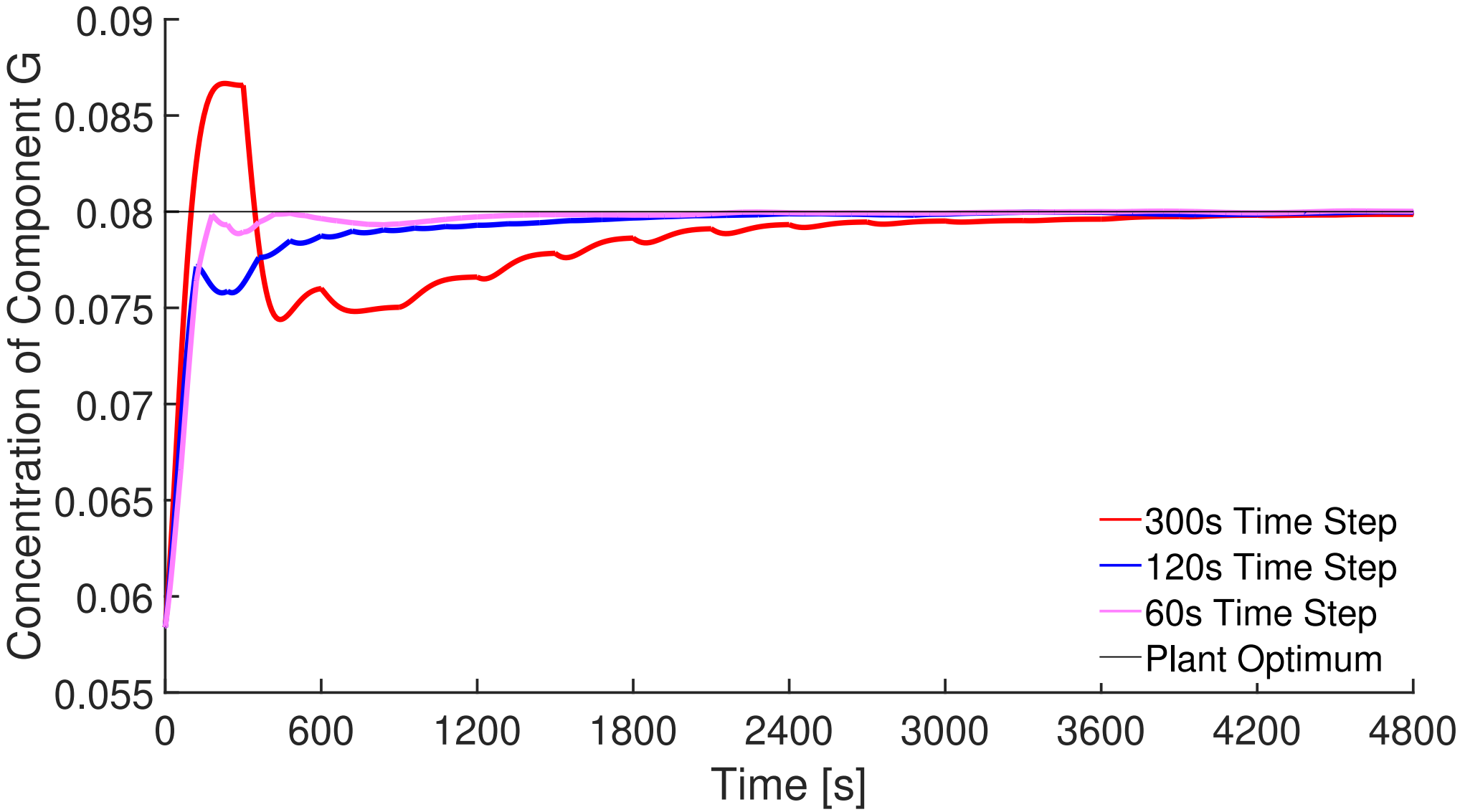

$\frac{\partial \phi_p}{\partial \boldsymbol{r}}$
Required by RTO scheme
$\boldsymbol{r}$
Controller
$\boldsymbol{u}_p$
Plant
ϕ_p
$\boldsymbol{y}_r$

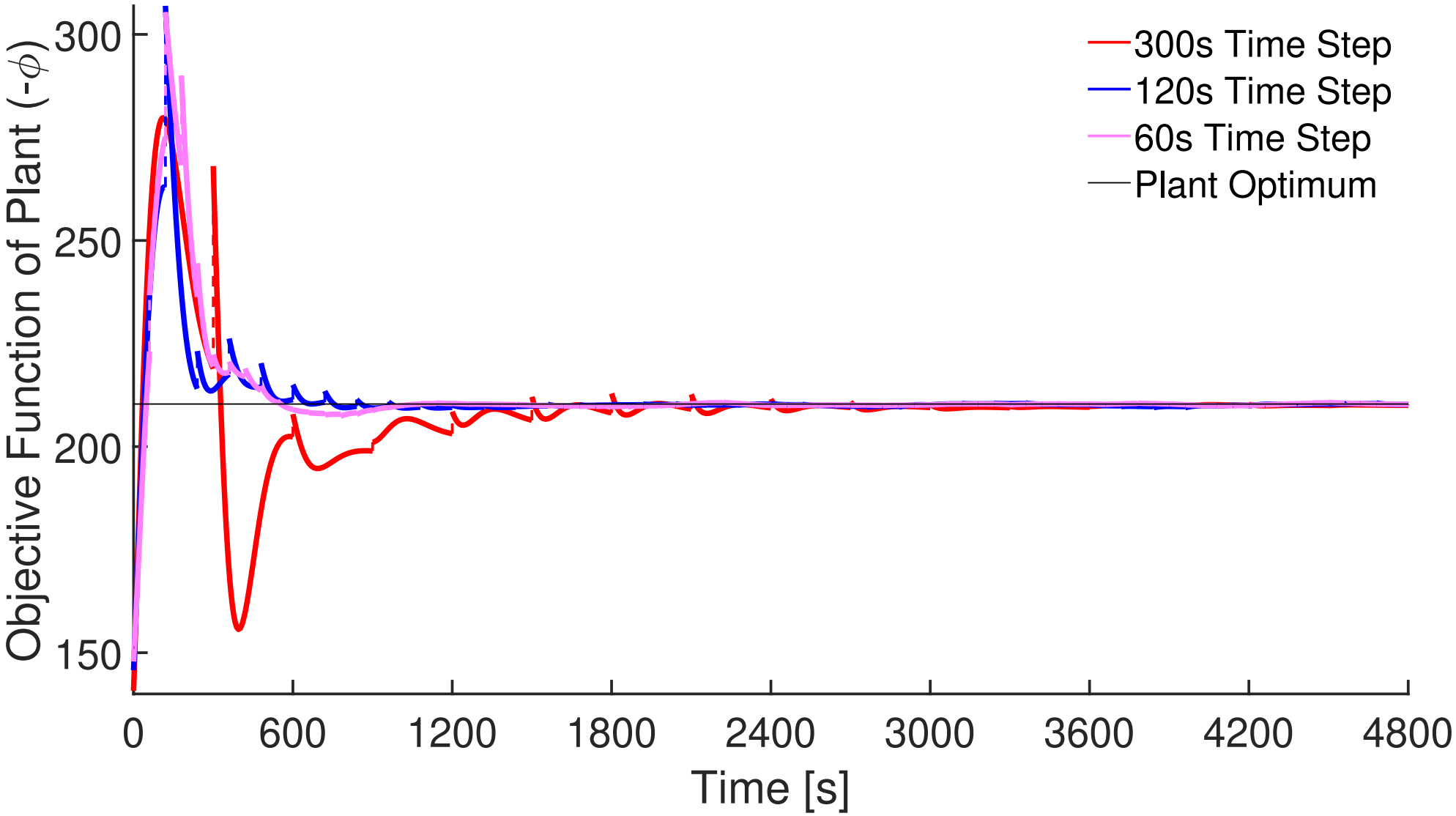

Current Operating Point
u_1
u_2
$\mathbb{D}$
$\mathbb{L}$

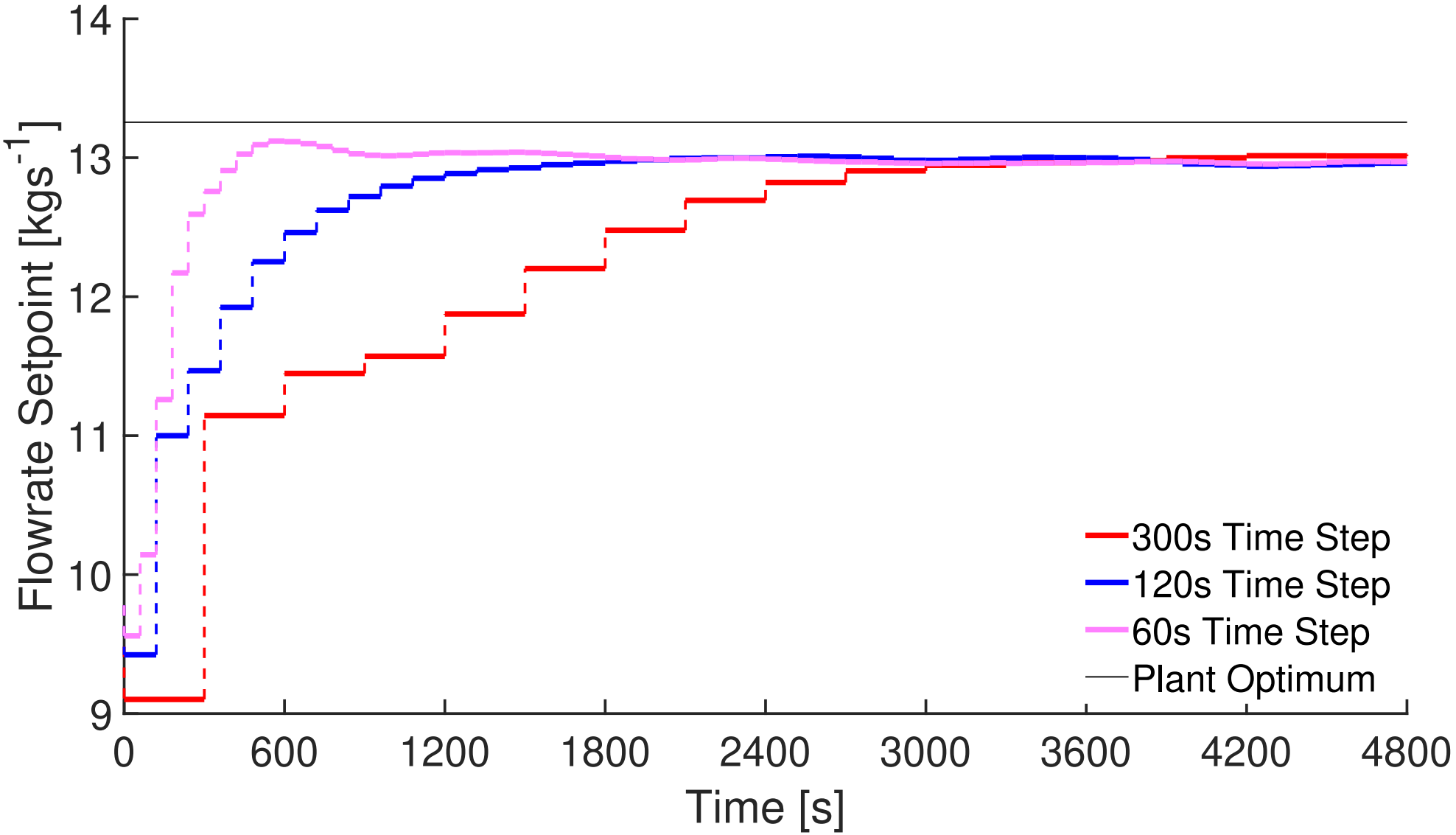

$\frac{\partial \phi_p}{\partial \boldsymbol{r}}$
Required by RTO scheme
$\boldsymbol{r}$
Controller
$\boldsymbol{u}_p$
Plant
ϕ_p
$\boldsymbol{y}_r$
$\frac{\partial \boldsymbol{u}_p}{\partial \boldsymbol{r}}$
$\frac{\partial \phi_p}{\partial \boldsymbol{u}_p}$

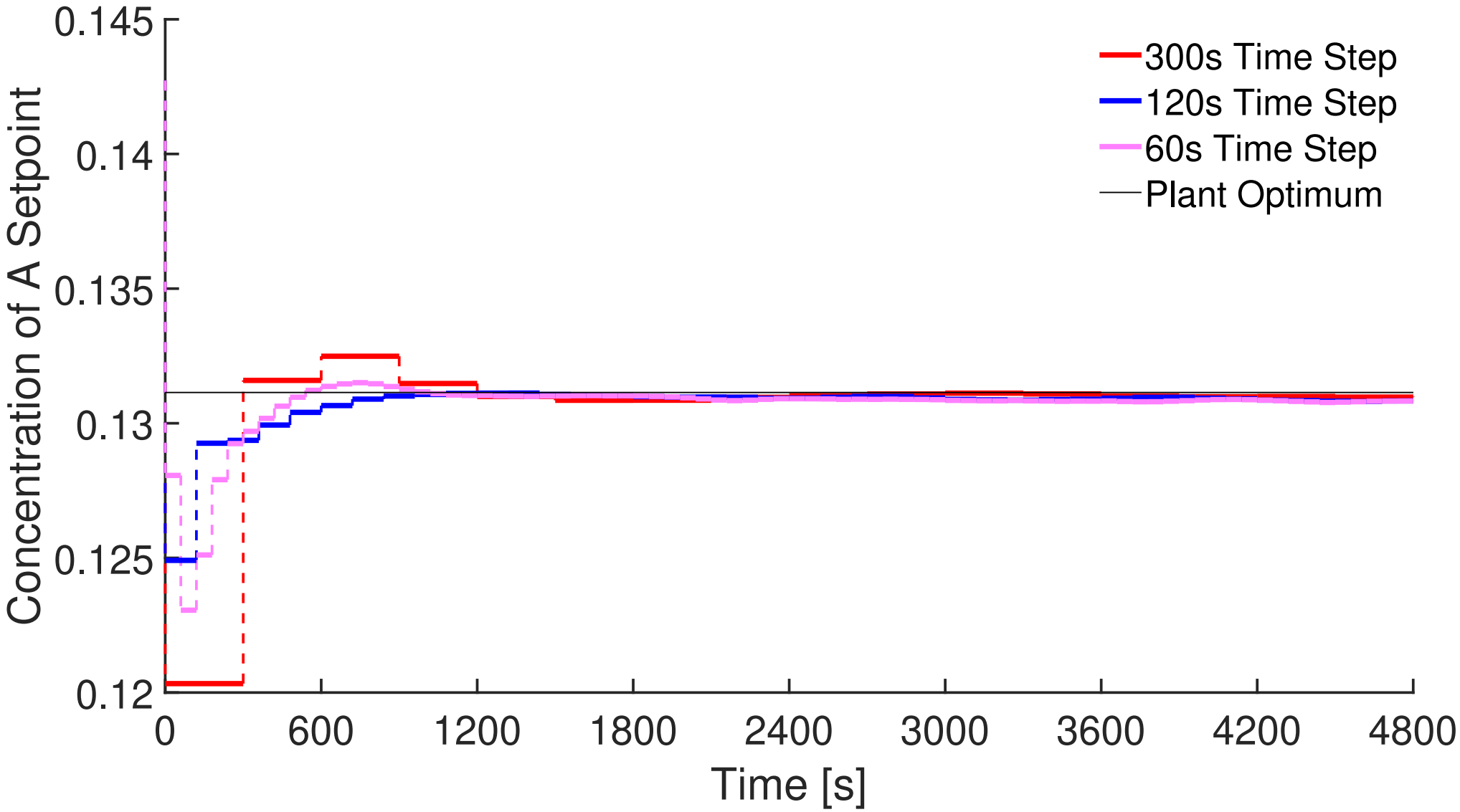

u_1
u_2
u_3
u_{n_u}
Model
r_1
r_2
r_{n_r}
Controller
$u_{p,1}$
$u_{p,2}$
$u_{p,3}$
u_{p,n_u}
$\mathbb{R}^{n_u}$
$\mathbb{R}^{n_r}$
$\mathbb{D}^{n_u}$


$\mathbb{R}^2$
$\mathbb{R}^1$
$\mathbb{D}^2$
u_1
u_2
r_1
$u_{p,1}$
$u_{p,2}$


0.09
0.085
0.08
0.075
0.07
0.065
0.06
0.055
Concentration of Component G
0
600
1200
1800
2400
3000
3600
4200
4800
Time [s]
300s Time Step
120s Time Step
60s Time Step
Plant Optimum


300s Time Step
120s Time Step
60s Time Step
Plant Optimum
Objective Function of Plant (-ϕ)
300
250
200
150
0
600
1200
1800
2400
3000
3600
4200
4800
Time [s]


14
13
12
11
10
9
Flowrate Setpoint [kgs^{-1}]
0
600
1200
1800
2400
3000
3600
4200
4800
Time [s]
300s Time Step
120s Time Step
60s Time Step
Plant Optimum


0.145
0.14
0.135
0.13
0.125
0.12
Concentration of A Setpoint
0
600
1200
1800
2400
3000
3600
4200
4800
Time [s]
300s Time Step
120s Time Step
60s Time Step
Plant Optimum

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jack Speakman: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization

Grégory François: Conceptualization, Writing - Review & Editing, Supervision

Real-Time Optimization via Modifier Adaptation of Closed-Loop Processes using Transient Measurements - Supplementary Material

1. Case Study 1 - Williams-Otto Reactor

As an initial example, a case study of a simple CSTR based on the Williams-Otto reactor (Williams and Otto, 1960) with structural plant-model mismatch is illustrated. The plant and model are discussed in detail below.

1.1. Open-loop Plant

The simulated plant follows a 3-reaction system, with two reactants (A and B), an intermediate component (C), two products (P and E), and a waste component (G). This system is described by the following set of reactions:

$$\begin{aligned} A + B &\xrightarrow{k_1} C, \\ B + C &\xrightarrow{k_2} P + E, \\ C + P &\xrightarrow{k_3} G. \end{aligned}$$

The reaction rate constants, k_j, can be calculated using the Arrhenius equation:

$$k_j = A_j e^{-E_j/T}, \tag{1}$$

where A_j and E_j are the pre-exponential factor and activation energy of reaction j respectively. It is assumed that the CSTR is isothermal at the chosen temperature, with ideal mixing. The dynamics of each component can be described using the following equation:

$$\frac{\partial X_i}{\partial t} = \frac{F_i}{M} + \sum_{j=1}^{3} \nu_{i,j} r_j - X_i \frac{F}{M}, \tag{2}$$

where $\nu_{i,j}$ is the stoichiometry of component i in reaction j, X_i is the mass fraction of component i in the reactor, M is the mass hold up of the CSTR, and r_j is the rate of reaction j, F_i is the inlet flow rate of species i and F is the total inlet (and outlet) flow rate the reactor. The reactions are assumed to be elementary, irreversible reactions, following the rate equation as follows:

$$r_j = k_j \prod_{i=1}^{6} X_i^{\nu_{i,j}^+}, \tag{3}$$

where $\nu_{i,j}^+$ are the positive stoichiometric coefficients. The open-loop inputs to this system have been chosen to be the inlet flow rates of reactants A and B, and the temperature of the system, $\boldsymbol{u}_p = [F_{A,p}, F_{B,p}, T_p]$. The parameters the of simulated plant are given in Table 1.

The objective function and constraints on the system can be written as:

$$\begin{aligned} \Phi(\boldsymbol{u}, \boldsymbol{y}) &= 76.23 F_A + 114.34 F_B \\ &\quad - 1143.38 X_P F - 25.92 X_E F, \end{aligned} \tag{4}$$

$$G(\boldsymbol{u}, \boldsymbol{y}) = X_G - 0.08 \leq 0. \tag{5}$$

Parameter	Value	Unit
A_1	1.6599×10^6	s^{-1}
A_2	7.2117×10^8	s^{-1}
A_3	2.6745×10^{12}	s^{-1}
E_1	6666.7	K
E_2	8333.3	K
E_3	11 111	K
M	2105	kg

Table 1: Simulated plant parameters

Parameter	Value	Unit
A_1^*	2.189×10^8	s^{-1}
A_2^*	4.310×10^{13}	s^{-1}
E_1^*	8075	K
E_2^*	12 400	K

Table 2: Simulated model nominal parameters

1.2. Open-loop Model

The model of this process, based on the model in (Roberts, 1979), is a steady-state model with the same inputs as the *open-loop* plant. This model has a significant degree of mismatch to the plant via a simplified, two reaction mechanism. This system does not contain the intermediate component C, and can be described as follows:

$$A + 2B \xrightarrow{k_1^*} P + E,$$
$$A + B + P \xrightarrow{k_2^*} G,$$

where k_j^* are the model reaction constants. The reaction rates are assumed to follow the Arrhenius equation $k_j^* = A_j^* e^{-E_j^*/T}$, with elementary reactions, similar to the plant. As this model is steady-state, the model can be solve via:

$$\frac{F_i}{M} + \sum_{j=1}^{2} \nu_{i,j} r_j - X_i \frac{F}{M} = 0, \tag{6}$$

for each component. The parameters for the model are summarized in Table 2, with the same mass hold up as the plant. NE is used for the gradient estimate of the plant, using the activation energies as the model parameters.

	Methanol	n-Propanol	Units
a_k	2.288	1.235	$\mathrm{mol\,L^{-1}}$
b_k	0.2685	0.271 36	
c_k	512.4	536.4	K
d_k	0.2453	0.2400	

Table 3: Molar volume relation coefficients

2. Case Study 2 - Distillation Column

As a more elaborate example with more complex dynamics to illustrate the proposed method on, a simulation of a distillation column was developed, based on (Diehl et al., 2001). This example is of a high purity binary distillation column, used in the separation of methanol and n-propanol. The column has 40 trays ($N = 40$), with the feed on tray 21 ($N_f = 21$), a partial reboiler, and total condenser, as illustrated in Figure 1. The open-loop system inputs are the reboiler heat duty, Q, and the reflux volumetric flow rate, L_{vol}. The control scheme investigated is based around controlling the temperatures on the key stages of trays 14 and 28.

The feed is subject to regular step changes in composition, and flow rate, due to a change in source of the feed stream, resulting is a changing set of optimal operating conditions. Therefore, the use of a transient method will be preferred to find the optimum, or close to, since convergence can be potentially achieved in a single iteration to steady state, avoiding multiple iterations to steady state while the plant optimal conditions has changed. The objective of the optimization is to maximize the operating profit, whilst meeting the specified distillate concentration:

2.1. Open-Loop Plant

The plant has been simulated using an equilibrium-based dynamic algebraic system (DAE). The trays are referred to by $l = 1, 2, ..., N$; the reboiler by $l = 0$; and the condenser by $l = N + 1$. The corresponding temperatures are denoted by $T_0, ..., T_{N+1}$. The composition of methanol in the liquid phase on tray l is denoted by x_l, and as the process is a binary system, the concentration of the n-propanol is denoted with $1 - x_l$ for $l = 0, ..., N + 1$.

Figure 2 illustrates the liquid and vapor fluxes from the trays, denoted by L_l and V_l respectively. The composition of the liquid streams equal the liquid phase compositions of their respective tray, x_l. Likewise the vapor streams have a composition equal to the respective tray vapor phase composition, y_l.

The pressure of the trays are assumed to be constant, as with the pressure drop between trays of 250Pa in the stripper and 190Pa in the rectifier. Therefore the pressure can be determined via $P_l = P_{l+1} + \Delta P_l$, with the pressure of the condenser defined as 93 900Pa. The volumetric holdup of the reboiler and condenser are assumed to be constant at 8.5L and 0.17L respectively. All molar holdups, n_l, are related to the volumetric holdups, n_l^v, via the molar volume $n_l^v = V_l^m n_l$. The molar volumes are calculated using the following relations:

$$V_l^m \coloneqq x_l V_{meth}^m + (1 - x_l) V_{prop}^m, \tag{7}$$

where V_{meth}^m and V_{prop}^m are the pure component molar volumes, given by:

$$V_k^m \coloneqq \frac{1}{a_k} (b_k)^{1+\left(1-\frac{T}{c_k}\right)^{d_k}}, \tag{8}$$

where k are the components. The coefficients are defined in Table 3.

The temperatures can be determined by assuming that the partial pressures of each component sum to the total pressure, for the reboiler and trays $l = 0, ..., N$:

$$P_l - P_{meth}^s(T_l) x_l - P_{prop}^s(T_l)(1 - x_l) = 0, \tag{9}$$

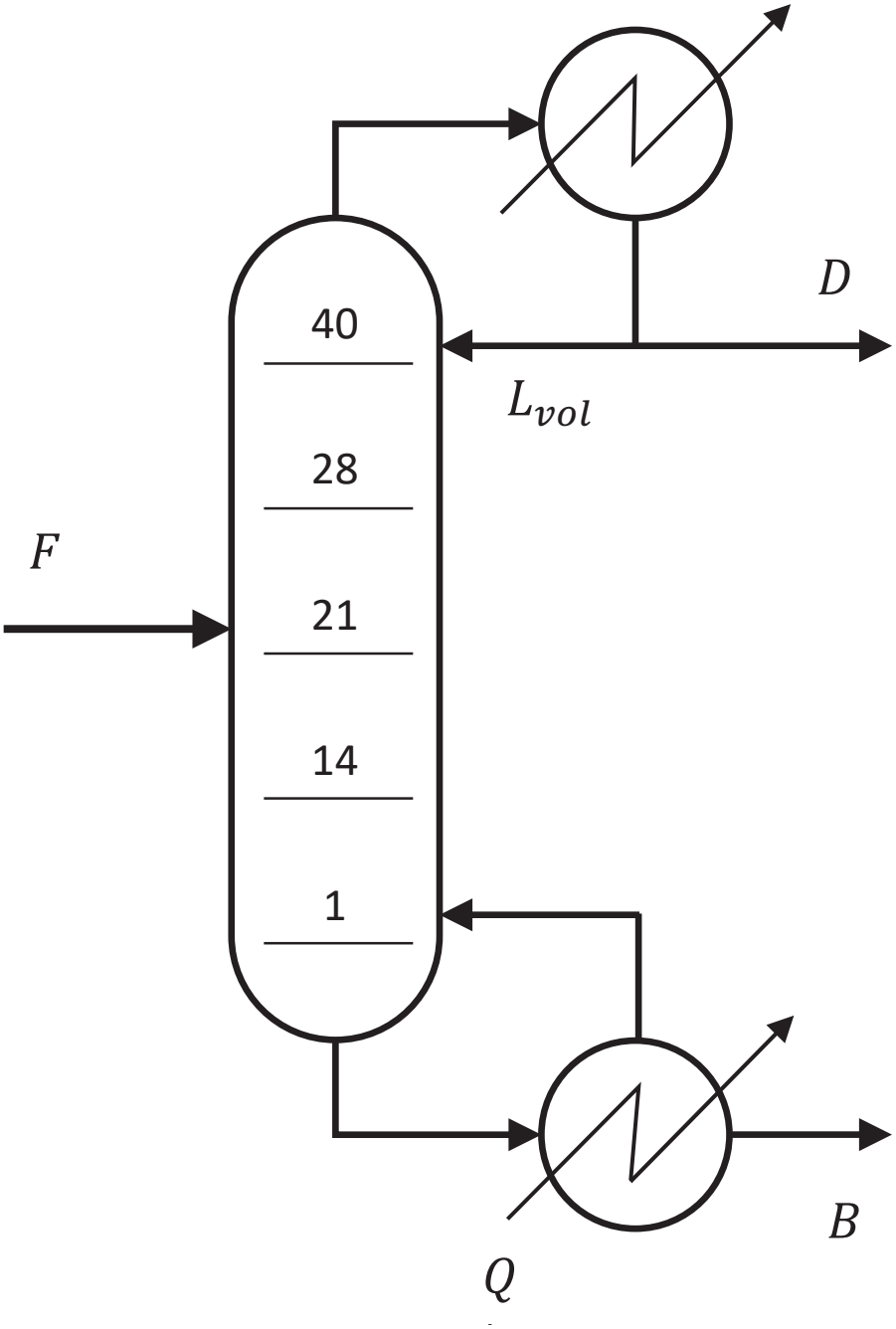


Figure 1: Flow sheet of the distillation column.

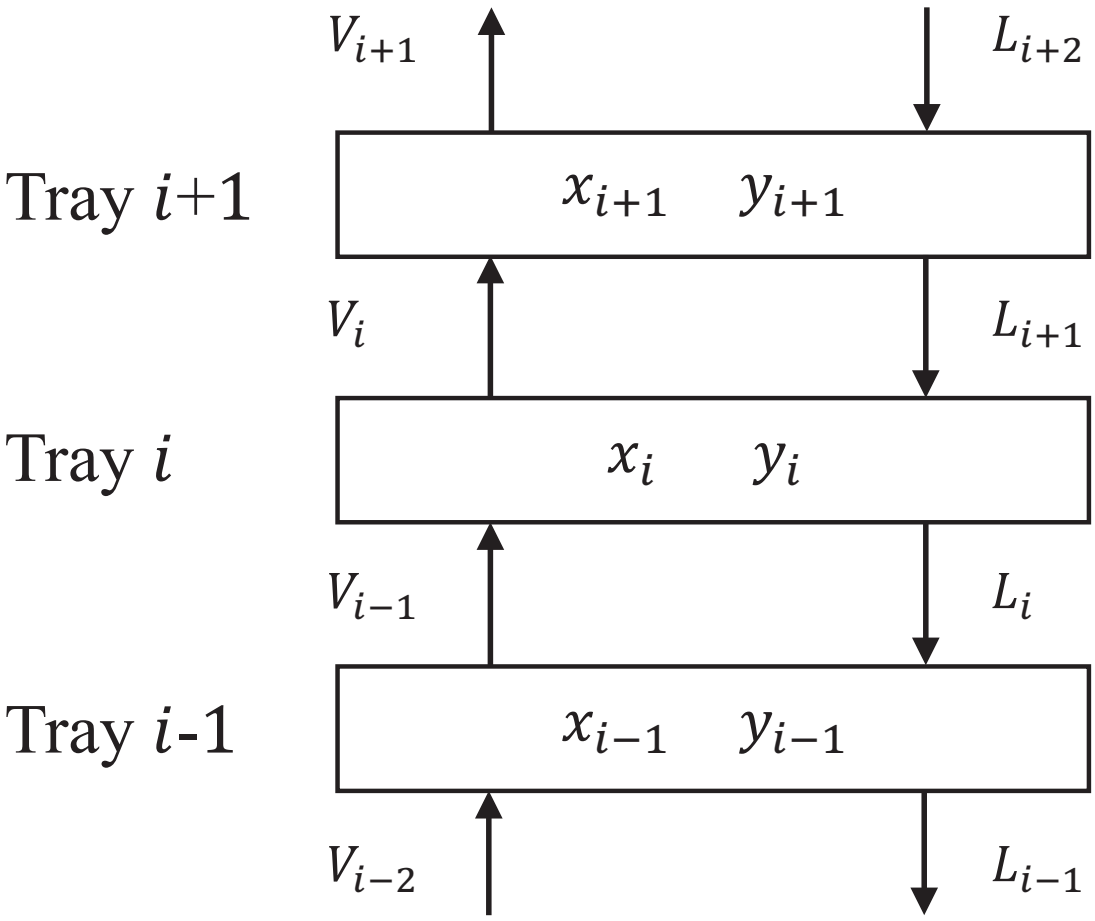


Figure 2: Flow sheet of vapor and liquid fluxes between trays.

	Methanol	n-Propanol	Units
A_k	23.48	22.437	
B_k	3626.6	3166.4	K
C_k	-34.29	-80.15	K

Table 4: Antoine component coefficients

with the temperature of the condenser fixed at 47.2°C, and where P_k^s is the pure component saturated pressure of component k at the stage temperature, determined via the Antoine equation:

$$P_k^s := e^{A_k - \frac{B_k}{T+C_k}}, \tag{10}$$

where the saturated pressure is in Pascals. The Antoine coefficients are given in Table 4.

The vapor compositions of each tray, y_l, can be determined by assuming they correspond to Murphrees plate efficiency equation. This equation introduces non-idealities of the tray by relating the actual tray composition to the ideal tray composition (determined via Raoult's law) and the incoming vapor composition from the tray below via an efficiency factor, α_l, for $l = 1, ..., N$. Murphrees plate equation can be given as:

$$y_l = \alpha_l \frac{P_1^s(T_l)}{P_l} x_l + (1 - \alpha_l) y_{l-1}, \tag{11}$$

assuming that the reboiler vapor liquid is at equilibrium, $y_0 = \frac{P_1^s(T_0)}{P_0} x_0$. The plate efficiencies in the stripper and rectifier are assumed to be 0.62 and 0.35 respectively. With all the states of the system defined, the DAE system can be described below. Beginning with the mass balances on each tray, for $l = 1, ..., N$:

$$\frac{\partial n_l}{\partial t} = V_{l-1} + L_{l+1} - V_l - L_l + F_l, \tag{12}$$

where F_l is the feed flow rate on to tray l. $F_{N_f} = F$, otherwise $F_l = 0$. Similarly for the component balance for $l = 1, ..., N$:

$$\begin{aligned} \frac{\partial X_l}{\partial t} n_l + \frac{\partial n_l}{\partial t} x_l =& V_{l-1} y_{l-1} + L_{l+1} x_{l+1} \\ & - V_l y_l - L_l x_l + F_l x_f. \end{aligned} \tag{13}$$

The mass balance of the reboiler and condenser can be defined separately as:

$$\frac{\partial n_0}{\partial t} = L_1 - V_0 - B, \tag{14}$$

$$\frac{\partial n_{N+1}}{\partial t} = V_N - D - L_{N+1}, \tag{15}$$

with the corresponding component balances:

$$\frac{\partial X_0}{\partial t} n_0 + \frac{\partial n_0}{\partial t} x_0 = L_1 x_1 - B x_0 - V_0 y_0, \tag{16}$$

$$\begin{aligned} \frac{\partial X_{N+1}}{\partial t} n_{N+1} + \frac{\partial n_{N+1}}{\partial t} x_{N+1} = \\ V_N y_N - D x_{N+1} - L_{N+1} x_{N+1}. \end{aligned} \tag{17}$$

With the volumetric holdup of the reboiler and condenser being fixed, two further equations can be defined for $l = 0, N + 1$:

$$\frac{\partial n_l^v}{\partial t} = V^m \frac{\partial n_l}{\partial t} + n_l \frac{\partial V^m}{\partial t} = 0. \tag{18}$$

	Methanol	n-Propanol	Units
$h_{1,k}$	18.31	31.92	K^{-1}
$h_{2,k}$	1.713×10^{-2}	4.49×10^{-2}	K^{-2}
$h_{3,k}$	6.299×10^{-5}	9.663×10^{-5}	K^{-3}
T_k^c	512.6	536.7	K
P_k^c	8.096×10^6	5.166×10^6	Pa
Ω_k	0.557	0.612	

Table 5: Enthalpy component coefficients

With the differential equations defined, the algebraic equations can be defined. For the liquid flow rate, a hydrodynamic equation can be introduced which relates the volumetric liquid flow rate to the volumetric holdup of the tray via the Francis weir formula for $l = 1, ...N$:

$$L_l V^m = W_l (n_l^v - n_l^{ref})^{\frac{3}{2}}, \tag{19}$$

where $W = 0.166\text{L}^{-0.5}\text{s}^{-1}$ and $n^{ref} = 0.155\text{L}$ are parameters of the formula. The liquid flow rate from the condenser can be determined from L_{vol}. Finally the vapor flow rate can be determined from the enthalpy balance in the reboiler:

$$\begin{aligned} \frac{\partial n_0}{\partial t} h_0^L + \frac{\partial h_0^L}{\partial t} n_0 =& Q - Q_{loss} \\ &+ L_1 h_1^L - V_0 h_0^V - B h_0^L, \end{aligned} \tag{20}$$

where $Q_{loss} = 0.51\text{kW}$ are the heat losses in the reboiler. Trays $l = 1, ...N$ can be similarly defined as:

$$\begin{aligned} \frac{\partial n_l}{\partial t} h_l^L + \frac{\partial h_l^L}{\partial t} n_l =& V_{l-1} y_{l-1} + L_{l+1} h_{l+1}^L \\ &- V_l h_l^V - L_l h_l^L + F_l h_f, \end{aligned} \tag{21}$$

where h^L and h^V are the enthalpies of the liquid and vapor streams respectively and the enthalpy differential can be defined explicitly w.r.t. the composition and temperature. The liquid and vapor enhthalpies can be calculated via the following equations:

$$h_l^L(X, T) \coloneqq x_l h_{meth}^L + (1 - x_l) h_{prop}^L, \tag{22}$$

$$h_l^V(X, T) \coloneqq y_l h_{meth}^V + (1 - y_l) h_{prop}^V, \tag{23}$$

where h_{meth}^L and h_{prop}^L are the pure component enthalpies, defined as:

$$\begin{aligned} h_k^L(T) \coloneqq & C \left(h_{1,k}(T - T_0) + \right. \\ & \left. h_{2,k}(T - T_0)^2 + h_{1,k}(T - T_0)^3 \right), \end{aligned} \tag{24}$$

$$\begin{aligned} h_k^V(T, P) \coloneqq & h_k^L + RT_k^c \sqrt{1 - P_k^r (T_k^r)^3} \\ & \left\{ a - bT_k^r + c(T_k^r)^7 + \Omega_k \left(d - eT_k^r + f(T_k^r)^7 \right) \right\}, \end{aligned} \tag{25}$$

where k is the two components, $T_k^r \coloneqq T/T_k^c$ is the reduced temperature, $P_k^r \coloneqq P/P_k^c$ is the reduced pressure and T_0 is the constant for the unit conversion from K to °C. The constants are defined in Tables 5 and 6.

To summarize, this 204 state DAE is formed of 82 differential states and 122 algebraic states. The differential states are formed of 42 composition states x_l for $l = 0, ..., N + 1$ given by Equations (13), (16) and (17), and 40 molar holdup states n_l for $l = 1, ..., N$ given by Equation (12). Whilst the algebraic states are formed of 41 volumetric fluxes V_l for $l = 0, ..., N$ given by Equations (20) and (21), 40 liquid fluxes L_l for $l = 1, ..., N$ given by Equation (19), and 41 temperatures T_l for $l = 0, ..., N$ given by Equation (9).

Constant	Value	Units
C	4.186	$\mathrm{J\,mol^{-1}}$
T_0	273.15	K
R	8.3147	$\mathrm{J\,mol^{-1}\,K^{-1}}$
a	6.096 48	
b	1.288 62	
c	1.016	
d	15.6875	
e	13.4721	
f	2.615	

Table 6: Enthalpy coefficients

2.2. Model

An equilibrium based model has been developed which is a simplification of the plant, the simplifications are summarized as follows

1. The molar-hold up of the trays are assumed to be constant
2. The reboiler and condenser molar holdup are assumed to be constant (with variable volumetric holdup)
3. The condenser is assumed to operate at the saturated liquid temperature
4. The model is steady-state

These simplification have the following impact on the model equation. Firstly, the molar holdup differential can be set to 0 for $l = 0, ..., N + 1$:

$$\frac{\partial n_l}{\partial t} = 0. \tag{26}$$

Therefore, the liquid flow rates can be directly calculated via the total mass holdup for the reboiler and trays, $l = 0, ..., N$:

$$L_l = V_{l-1} + L_{l+1} - V_l + F_l. \tag{27}$$

The resulting model has 125 states, with no molar holdup and liquid flux states, but an additional temperature state for the condenser (i.e. Equation (9) applies for $l = N + 1$). In addition to the simplifying discussed above, the Murphree plate efficiencies are assumed to be different between the plant and model with efficiencies of 0.68 and 0.32 in the stripper and rectifier respectively. The feed is assumed to be fixed with a flow rate of $14\mathrm{L\,h^{-1}}$ and a concentration of 0.32 at 71°C. This results in structural mismatch between the open-loop plant and model. The parameters used by NE are the plate efficiencies, feed flow rate and feed composition.

References

Diehl, M., Uslu, I., Findeisen, R., Schwarzkopf, S., Allgöwer, F., Bock, H.G., Bürner, T., Gilles, E.D., Kienle, A., Schlöder, J.P., Stein, E., 2001. Real-Time Optimization for Large Scale Processes: Nonlinear Model Predictive Control of a High Purity Distillation Column, in: Online Optimization of Large Scale Systems. Springer Berlin Heidelberg, pp. 363–383.

Roberts, P.D., 1979. An algorithm for steady-state system optimization and parameter estimation. International Journal of Systems Science 10, 719–734.

Williams, T.J., Otto, R.E., 1960. A generalized chemical processing model for the investigation of computer control. Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics 79, 458–473.